

공학석사 학위논문

Stereo Vision System을 이용한  
휴머노이드 로봇의 12자유도 팔 제어시스템 개발

Development of Control System for a Humanoid Robot Arm with 12 DOF  
using the Stereo Vision System

지도교수 최 형 식

2007년 2월

한국해양대학교 대학원

기 계 공 학 과

김 병 국

# 목 차

Abstract

기호설명

그림목차

표목차

제 1 장. 서 론 ..... 1

제 2 장. 로봇의 전체시스템 구성 ..... 3

2.1 로봇의 전체시스템 구성 ..... 3

2.2 로봇의 제어시스템 구성 ..... 6

제 3 장. 로봇상체의 구조 및 기구학적 해석 ..... 8

3.1 머리부의 구조 ..... 8

3.2 양팔부의 구조 ..... 10

3.3 순기구학 해석 ..... 17

3.4 역기구학 해석 ..... 23

제 4 장. 관절제어기의 구성 ..... 26

4.1 관절제어기의 하드웨어적 구성 ..... 26

4.2 모터제어 알고리즘 ..... 27

4.3 CAN 통신을 이용한 제어기의 네트워크 구성 ..... 29

|   |    |
|---|----|
| 제 5 장. Stereo Vision System의 구성 .....         | 32 |
| 5.1 Vision System의 구성 및 사양 .....              | 32 |
| 5.2 OpenCV를 이용한 영상처리 및 물체거리 산출 .....          | 34 |
| 제 6 장. x86기반의 보행로봇 제어기 .....                  | 37 |
| 6.1 Win32 Multi-thread/Message 기반의 프로그램 ..... | 37 |
| 6.2 프로그램의 구성 .....                            | 39 |
| 제 7 장. 3D 시뮬레이션 .....                         | 42 |
| 7.1 시뮬레이터의 구성 .....                           | 43 |
| 7.2 물리엔진의 소개 .....                            | 44 |
| 7.3 로봇 모델링 .....                              | 45 |
| 7.4 관절제어시스템 구성 .....                          | 46 |
| 제 8 장. 실험 및 고찰 .....                          | 48 |
| 8.1 물체검출 및 거리측정 실험 .....                      | 48 |
| 8.2 팔의 모션제어 .....                             | 51 |
| 제 9 장. 결 론 .....                              | 54 |

## 참고문헌

# Development of Control System for a Humanoid Robot Arm with 12 DOF using the Stereo Vision System

Kim, Byung Gook

Department of Mechanical Engineering  
Graduate School, Korea Maritime University

## Abstract

In this thesis, we developed a stereo vision system using CCD cameras to coordinate the motions of two arms of a humanoid robot with 27 degree of freedom(d.o.f). To do this, a closed-form solution of the inverse kinematics for the end-point of two arms with 12 d.o.f. was obtained and coordinates of the specific object were computed for the vision system. Using mapping coordinates computed by the relation between the vision system and the end-point of the robot arm, a visual-servoing system for the humanoid robot was developed.

In addition to this, a 3D graphical simulator was developed using the physical software-engine to simulate the two-arm motion coordination system before performing experiments for the system.

Some experiments to verify the motion coordination of the visual servoing system were performed. Through the experiments, a successful result that the gripper of the arm catches the object was achieved.

## 기 호 설 명

|                 |                     |
|-----------------|---------------------|
| $l_1, l_2, l_3$ | 4절링크의 링크길이          |
| $d$             | 4절링크의 미끄럼변위         |
| $\theta$        | 4절링크의 관절각           |
| $A_i$           | 각 관절의 동차 변환 행렬      |
| $T$             | 기구의 동차 변환 행렬        |
| $P_{head}$      | 머리의 말단장치 위치벡터       |
| $P$             | 팔의 말단장치 위치벡터        |
| $\theta_1$      | 어깨관절의 yaw 각         |
| $\theta_2$      | 어깨관절의 pitch 각       |
| $\theta_3$      | 팔꿈치관절의 pitch 각      |
| $\theta_4$      | 손목관절의 pitch 각       |
| $\theta_5$      | 손목관절의 yaw 각         |
| $f$             | 카메라 렌즈와 이미지 센서간의 거리 |
| $D$             | 이미지 센서상의 거리 차       |
| $B$             | 두 카메라간의 거리          |
| $Z$             | 카메라에서 물체까지의 거리      |
| $W$             | 이미지센서의 폭            |
| $w$             | 이미지의 폭              |
| $c_i$           | $\cos\theta_i$      |
| $s_i$           | $\sin\theta_i$      |

## 그 림 목 차

|  |    |
|--|----|
| Fig. 2.1 Four-bar link .....                                   | 3  |
| Fig. 2.2 Specification of the humanoid robot .....             | 4  |
| Fig. 2.3 Total control system of the humanoid robot .....      | 7  |
|  |    |
| Fig. 3.1 Front and side view of the head .....                 | 9  |
| Fig. 3.2 Rotation range of the head .....                      | 9  |
| Fig. 3.3 3D model of the shoulder joint actuator .....         | 10 |
| Fig. 3.4 Rotation range of the shoulder pitch joint .....      | 10 |
| Fig. 3.5 Shoulder joint .....                                  | 10 |
| Fig. 3.6 3D model of the elbow joint actuator .....            | 11 |
| Fig. 3.7 Rotation range of the elbow pitch joint .....         | 11 |
| Fig. 3.8 Elbow joint .....                                     | 11 |
| Fig. 3.9 3D model of the wrist joint actuator .....            | 12 |
| Fig. 3.10 Rotation range of the wrist pitch joint .....        | 12 |
| Fig. 3.11 Wrist joint .....                                    | 12 |
| Fig. 3.12 Four-bar link modeling .....                         | 14 |
| Fig. 3.13 Four-bar link modeling for the shoulder .....        | 15 |
| Fig. 3.14 Four-bar link modeling for the elbow .....           | 16 |
| Fig. 3.15 Four-bar link modeling for the wrist .....           | 16 |
| Fig. 3.16 Coordinates of the head .....                        | 18 |
| Fig. 3.17 Coordinates of the arm .....                         | 20 |
| Fig. 3.18 Projection to the $x_0y_0$ plane .....               | 23 |
| Fig. 3.19 Projection to the plane formed with link 2,3,4 ..... | 24 |
|  |    |
| Fig. 4.1 Joint controller .....                                | 26 |

|   |    |
|---|----|
| Fig. 4.2 Flow chart of the timer ISR .....                                      | 27 |
| Fig. 4.3 Block diagram of the PID controller .....                              | 28 |
| Fig. 4.4 Velocity profile .....   | 28 |
| Fig. 4.5 Message frame of CAN 2.0a .....  | 29 |
|   |    |
| Fig. 5.1 Stereo vision system .....   | 33 |
| Fig. 5.2 Internal structure of the image grabber 'Morphis' .....                | 34 |
| Fig. 5.3 Theory for computing the distance using the stereo vision system ..... | 35 |
|   |    |
| Fig. 6.1 General idea for the multi-threaded programming .....                  | 38 |
| Fig. 6.2 Block diagram of the main controller program .....                     | 40 |
|   |    |
| Fig. 7.1 Block diagram of the 3D simulator .....                                | 42 |
| Fig. 7.2 3D graphical simulator .....   | 43 |
| Fig. 7.3 Joint types of ODE .....   | 45 |
|   |    |
| Fig. 8.1 Image processing of the vision system .....                            | 48 |
| Fig. 8.2 Tracking experiment of the head .....                                  | 49 |
| Fig. 8.3 View of tracking control .....   | 50 |
| Fig. 8.4 Result of computing distance using the stereo vision system .....      | 51 |
| Fig. 8.5 Flow chart of visual servoing .....                                    | 52 |
| Fig. 8.6 Motion of the arm .....  | 53 |

## 표 목 차

|   |    |
|---|----|
| Table 2.1 Specification of the humanoid robot .....               | 5  |
| Table 2.2 Degree of freedom of the humanoid robot .....           | 5  |
| Table 3.1 Specification of the joint actuator for head part ..... | 8  |
| Table 3.2 Specification of the joint actuator for arm part .....  | 13 |
| Table 3.3 Length of the link .....                                | 15 |
| Table 3.4 Link parameter for head part .....                      | 18 |
| Table 3.5 Link parameter for arm part .....                       | 21 |
| Table 4.1 Formed CAN ID .....                                     | 30 |
| Table 4.2 Contents of CAN .....                                   | 31 |
| Table 5.1 Specification of the stereo vision system .....         | 33 |
| Table 6.1 Windows Message .....                                   | 39 |



# 제 1 장 서 론

현재 인간을 대신하여 작업을 하는 산업용 로봇은 인간의 생활이나 공장자동화에 지대한 영향을 미치고 있으며, 두 대의 로봇이 협조 작업을 하는 등 고도의 로봇제어기술이 개발되고 있다. 로봇은 단순반복적인 작업에서부터 자동차 조립과 같이 정밀하고 어려운 작업, 인간이 갈 수 없는 극한지역에서의 작업까지 여러 가지 분야에 다양한 형태로 적용되고 있다. 이렇게 유용한 로봇을 인간이 생활하는 공간에 가까이 두어 보다 편리한 생활환경을 만들기 위한 노력이 이어지고 있는데, 그 중 한 가지가 두 발로 걸어 다니는 이족보행로봇, 즉 휴머노이드(Humanoid) 로봇이다.

현재 휴머노이드 연구에 있어서 선도적인 위치를 점하고 있는 나라는 일본으로써, 여러 기업과 연구소 및 대학에서 폭넓은 휴머노이드 연구가 활발히 진행되고 있으며 대표적인 예가 2000년에 혼다(Honda)가 발표한 ‘ASIMO’ 와 소니(Sony)의 ‘AIBO’ 이다[1~2]. 최근 국내에서도 연구가 활발히 진행되고 있는데, 한국과학기술원의 ‘HUBO’ 와 정보통신부에서 개발한 ‘마루’ 와 ‘아루’, 그리고 본 논문에서 다루는 한국해양대학교의 ‘KUBIR’ 가 대표적인 사례이다[3].

인간의 관절구조를 모방한 휴머노이드는 상용화가 2020년 이후로 예상되고 있으며, 로봇은 기계, 전기전자, 프로그래밍 등과 같은 다양한 분야의 학문과 기술이 종합적으로 접목된 것으로써 개발에 관한 상징성이 매우 크고 인간과 동일한 외형을 갖추고 있어 친화적인 서비스를 제공할 수 있다는 측면에서 로봇개발이 세계적으로 가속화되고 있다.

또한 인간에게 좀 더 친화적인 서비스를 제공하기 위하여 단순히 동작이 프로그램되어 움직이는 기계가 아닌 감지 및 인지능력, 인간과의 상호작용 능력, 자율주행 및 조작능력 등을 가지는 지능형 로봇으로의 개발이 활발한 추세이며 이

러한 기술을 위해 선행되어야할 감지 및 인지기술을 위한 시각, 촉각, 청각 센서의 개발도 활발하게 이뤄지고 있다.

로봇이 감지할 수 있는 감각 중 시각정보를 이용하여 로봇의 관절을 제어하는 것을 비주얼 서보잉(visual servoing)[4]이라고 하며, 산업용 로봇에 많이 사용되어졌다. 휴머노이드 로봇의 경우 인간을 모방하였기 때문에 시각정보를 이용하는 것은 필수적인 기능이며, 비주얼 서보잉을 휴머노이드 로봇에 적용하려는 시도가 점차 증가하고 있으며[5] 앞서 언급한 ‘ASIMO’의 경우, 로봇이 전방에 서 있는 사람을 인식하고 반응할 수 있을 정도로의 기술이 개발된 상태이다[2].

본 논문에서는 27자유도를 가지는 이족보행로봇 ‘KUBIR(Korea Maritime University Biped Robot)’에 2대의 CCD 카메라와 이미지그래버를 이용하여 스테레오 비전시스템을 구축하였으며 임베디드 소형 컴퓨터를 이용하여 로봇 주제어시스템을 개발하였다. 또한 상체 12자유도의 양팔에 대하여, 말단장치좌표의 제어를 위한 팔의 관절각을 변화시킬 수 있는 역기구학의 닫힌 해(closed-form)를 구했으며 비전시스템을 이용하여 산출된 특정물체의 위치를 말단장치좌표와 매핑(mapping)하여 비주얼서보잉을 구현하였고 실제 구동실험에 앞서 3D 그래픽 시뮬레이터를 개발하여 연구된 제어이론을 검증해보았다.

본 논문의 2장에서는 로봇의 전체적 시스템 구조, 특히 로봇의 제어부에 관하여 소개하였고, 3장에서는 로봇의 기구적 구조를 설명하였고 기구학 해석을 통하여 관절각 제어를 위한 역기구학의 닫힌 해를 제시하였다. 그리고 4장에서는 관절제어기와 제어기 네트워크에 대해 소개하였고, 5장에서는 스테레오 비전 시스템에 관하여 설명하였다. 6장에서는 주제어기 구성 및 프로그램, 7장에서는 개발된 시뮬레이터에 대해 논하였다. 8장에서는 개발된 전체 제어시스템에 대한 실험내용과 결과를 보였으며, 마지막 9장에서는 결론과 향후과제에 대해 논하였다.

## 제 2 장 로봇의 전체시스템 구성

### 2.1 로봇의 전체시스템 구성

개발된 휴머노이드 로봇은 전체 키 160cm, 몸무게 75kg이며, 한 다리 당 6자유도, 한 팔당 6자유도, 머리 2자유도, 허리 1자유도를 가져 총 27자유도를 갖는 성인 크기의 로봇이다. 로봇의 머리에는 눈에 해당하는 CCD 카메라 2대가 장착되어 있으며 머리는 yaw와 pitch운동이 가능하다. 팔의 6자유도는 어깨에서부터 yaw-pitch-pitch-pitch-roll의 구조로 되어 있고 말단장치로 사람의 손 역할을 하는 그리퍼(gripper)가 장착되어 있어 물건을 잡을 수 있는 구조이다. 다리의 6자유도는 yaw-roll-pitch-pitch-pitch-roll의 구조로 되어 있어 인간과 유사한 보행을 할 수 있는 구조로 설계되어 있으며 또한 팔부분과 다리부분을 연결하는 허리관절은 로봇의 보행 시 무게중심을 조절하는 역할을 하게 된다. 다리와 팔의 12자유도 중 pitch 운동을 하는 6개의 관절은 모터 직결 구조가 아닌 Fig. 2.1과 같이 볼스크류(ball-screw)를 응용하여 직선운동을 회전운동으로 바꾸어 큰 토크(torque)를 낼 수 있는 4절링크 구조로 되어 있다.

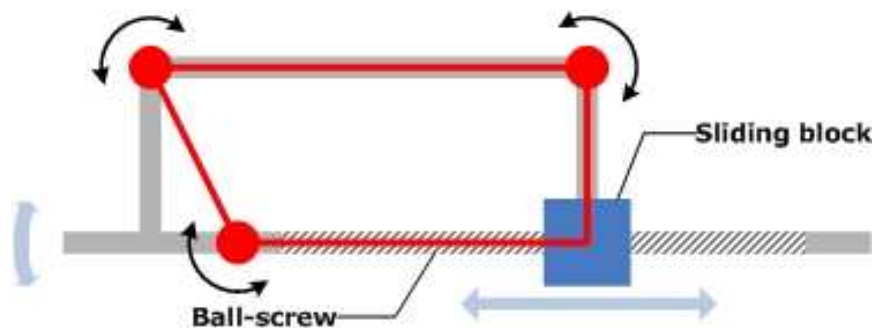


Fig. 2.1 Four-bar link

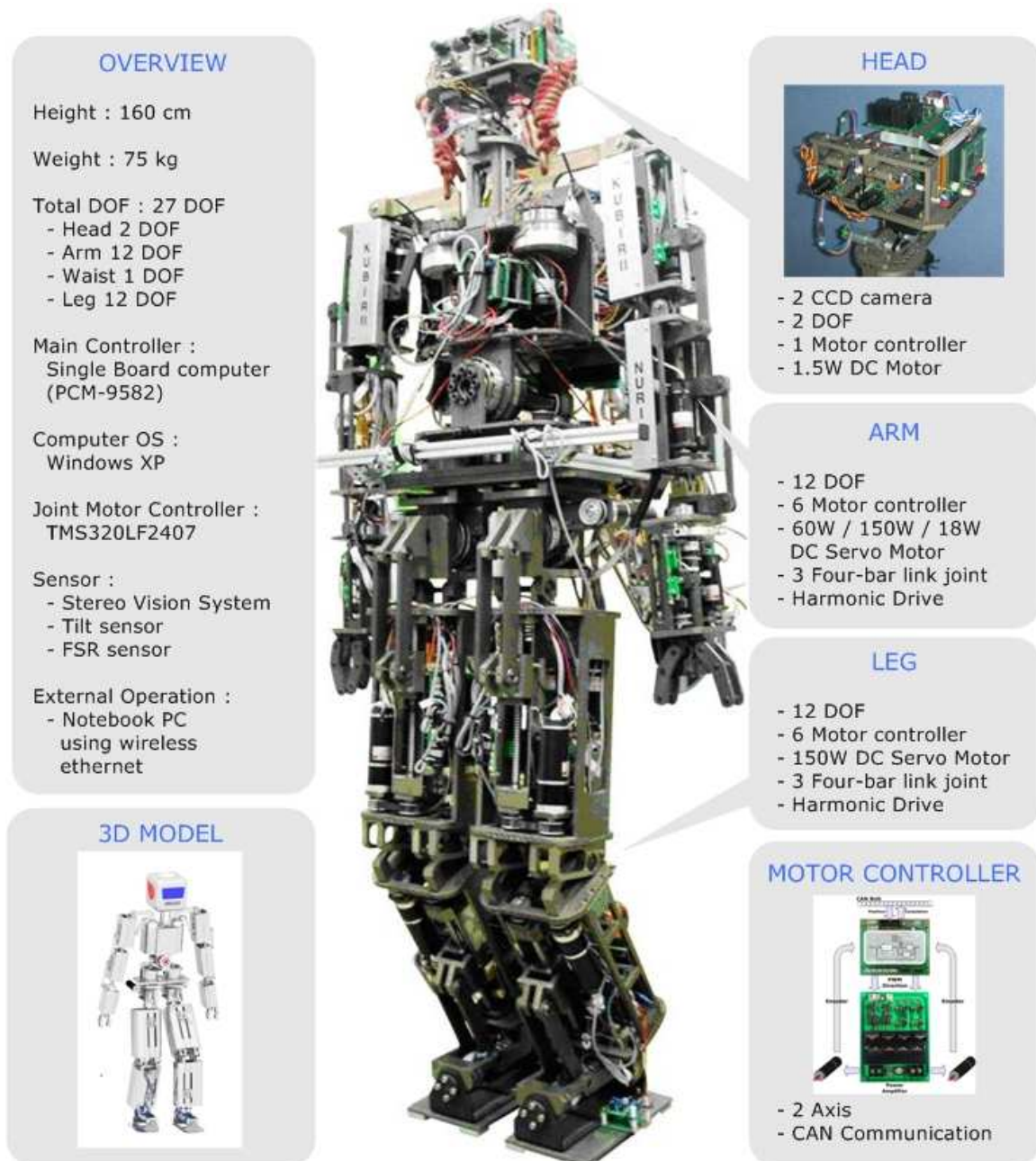


Fig. 2.2 Specification of the humanoid robot

**Table 2.1 Specification of the humanoid robot**

|                  |  |  |
|------------------|--|--|
| Height           | 160 [cm]   |  |
| Weight           | 75 [kg]  |  |
| Actuator         | Arm  | DC Servo motor +<br>Harmonic speed reducer |
|                  | Leg  | DC Servo motor +<br>Ball Screw             |
|                  | Waist  |  |
|                  | Head   | DC motor                                   |
| Control unit     | Main controller  | Embedded<br>Single-board Computer          |
|                  | Joint controller   | TMS320LF2407 + Driver                      |
| Power capacity   | 24V / 50AH   |  |
| Sensory device   | 2 CCD camera / Image grabber<br>Tilt sensor, FSR sensor<br>Magnetic sensor, Proximity sensor |  |
| Operation device | Notebook PC with wireless LAN  |  |

**Table 2.2 Degree of freedom of the humanoid robot**

|       |        |  |
|-------|--------|--|
| Head  | 2 DOF  |  |
| Arm   | Left   | 6 DOF<br>(shoulder 2 + elbow 1 + wrist 1 + hand 1) |
|       | Right  | 6 DOF<br>(shoulder 2 + elbow 1 + wrist 1 + hand 1) |
| Waist | 1 DOF  |  |
| Leg   | Left   | 6 DOF<br>(pelvis 3 + knee 1 + ankle 2)             |
|       | Right  | 6 DOF<br>(pelvis 3 + knee 1 + ankle 2)             |
| Total | 27 DOF |  |

## 2.2 로봇의 제어시스템 구성

개발된 로봇은 27개의 관절을 가지고 있으며 관절을 움직이기 위해 사용된 모터는 27개이다. Fig. 2.3 은 로봇의 전체 제어시스템을 나타낸 것이다. 모터 2개는 하나의 컨트롤러(controller)와 드라이버로 구성된 관절제어기로 제어되고 각각의 관절제어기는 CAN(Controller Area Network)으로 연결되어 있으며 관절제어기의 상위에는 로봇의 주제어기인 임베디드용 소형 컴퓨터가 있어 각 관절의 구성조건에 따른 연산을 하고 관절제어기로 제어명령을 내리게 된다. 이로써 27개의 관절을 가진 이족보행로봇의 전체 관절이 제어되어 로봇의 움직임을 만들 수 있게 된다.

개발에 사용된 컴퓨터에는 무선 랜카드가 장착되어 있어 외부에서 로봇으로 명령을 내릴 수 있으며 반대로 관절각 및 에러 등과 같은 로봇의 현재 상태를 외부로 알릴 수 있다. 사용된 무선이더넷 네트워크는 애드혹(ad-hoc)으로써 애드혹은 네트워크의 구성 및 유지를 위해 기지국이나 액세스포인트(access point)와 같은 기반 네트워크 장치가 필요하지 않다는 장점이 있다. 따라서 조종기 역할을 하는 노트북의 무선랜과 로봇 주제어기의 무선랜이 중계기 없이 바로 연결되어 통신이 가능하다.

또한 로봇의 머리 부분에 설치된 2대의 CCD 카메라로부터 들어오는 아날로그 영상신호를 획득하여 이진화하기 위해 주제어기에는 이미지그래버가 설치되어 있으며 로봇은 영상처리를 통해 입력된 영상에서 특정 물체를 검출하여 관절제어에 적용할 수 있도록 되어 있다.

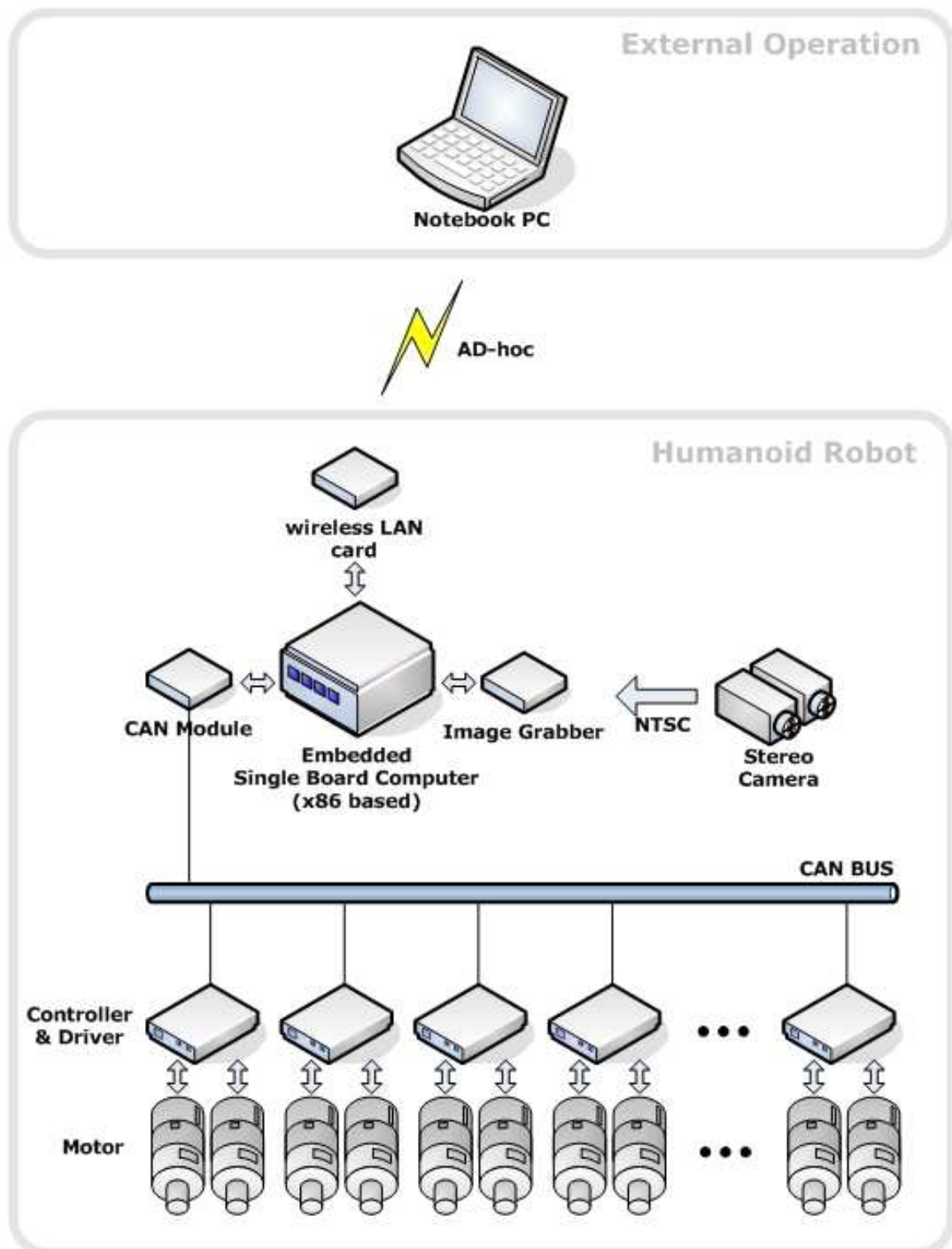


Fig. 2.3 Total control system of the humanoid robot

## 제 3 장 로봇상체의 구조 및 기구학적 해석

본 연구에서 다루는 이족보행로봇의 상체는 전체 14개의 모터로 구동되는 시스템으로써 6개의 직결구조 관절과 8개의 4절링크 구조의 관절로 구성되어 있다. 로봇의 관절은 머리부분 2자유도와 팔부분 12자유도를 가지고 있으며, 머리부분은 yaw운동과 pitch운동이 가능하며 감속기와 풀리(pulley)를 이용하여 구동기와 직결되어 있다. 팔부분에서는, 어깨부분의 yaw축과 손목부분의 roll축은 감속기를 이용하여 구동기와 직결되어 있으나, 어깨, 팔꿈치, 손목의 pitch축과 말단장치인 그리퍼(gripper)는 4절링크 구조에 볼스크류를 적용한 형태로 토크의 전달력과 강성을 크게 하고, 제어정밀도를 높이는 구조로 되어있다.

### 3.1 머리부의 구조

휴머노이드 로봇의 머리는 Fig. 3.1 과 같이 2자유도를 가지고 있으며, 인간의 눈과 같은 역할을 하는 2대의 CCD 카메라가 장착되어 있어 로봇 전방의 물체를 인식하는 역할을 한다. Table 3.1 과 같이 2개의 모터가 감속기와 풀리를 이용, 직결되어 관절을 구성하고 있으며 Fig. 3.2 와 같이 카메라가 바라보는 각도를 기준으로 상하 90°와 좌우 90°의 운동범위를 가지고 있어 카메라의 입사각을 고려하면 카메라가 볼 수 있는 시야는 로봇 전방 전체이다.

Table 3.1 Specification of the joint actuator for head part

| 축 | 운동    | 모터출력[W] | 풀리비     | 감속비     |
|---|-------|---------|---------|---------|
| 0 | yaw   | 1.5     | 14 : 40 | 1 : 104 |
| 1 | pitch | 1.5     | 14 : 40 | 1 : 104 |



Fig. 3.1 은 실제 머리의 모습으로써 머리에는 2대의 CCD카메라와 모터, 그리고 모터를 제어하기위한 관절제어기와 드라이버가 장착되어 있다.

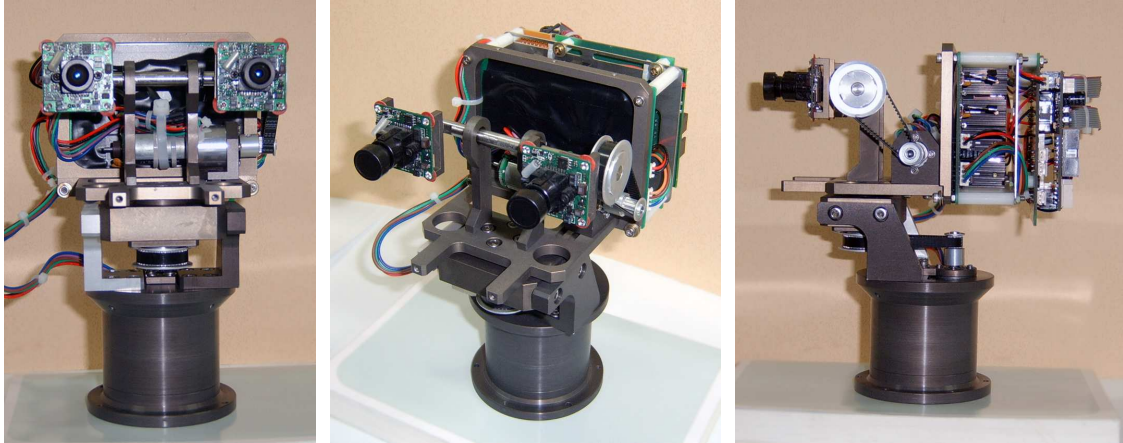


Fig. 3.1 Front and side view of the head

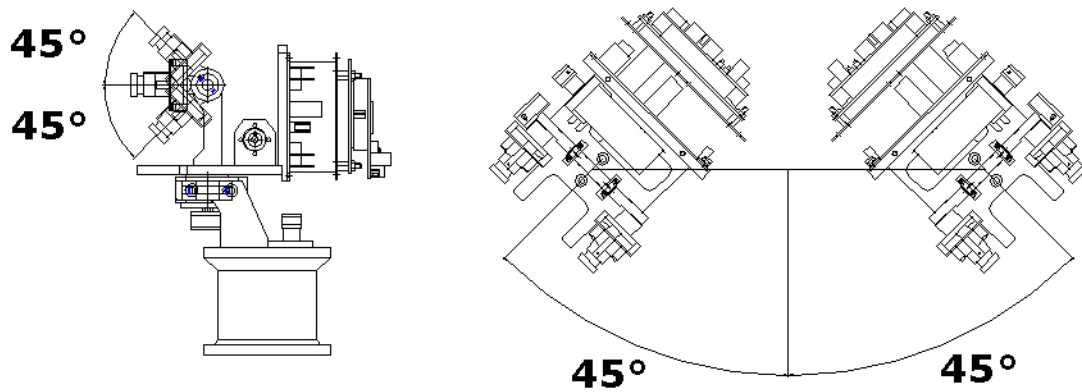


Fig. 3.2 Rotation range of the head

### 3.2 양팔부의 구조

로봇의 양팔은 총 12자유도를 가지고 있으며 4개의 직결구조 yaw, roll축과 6개의 4절링크구조 pitch축, 그리고 말단장치인 그리퍼 2축으로 구성되어 있다.

Fig. 3.3 은 어깨부분의 관절구조로써 직결구조의 yaw축이 팔 전체를 회전할 수 있도록 되어 있으며 첫 번째 pitch축이 위치하고 있다. 어깨부분의 yaw운동은 로봇의 앞쪽으로 90°의 범위로 이루어지며 pitch축은 Fig. 3.4 와 같이 로봇 몸체로부터 바깥쪽으로 49°의 운동범위를 갖는다.

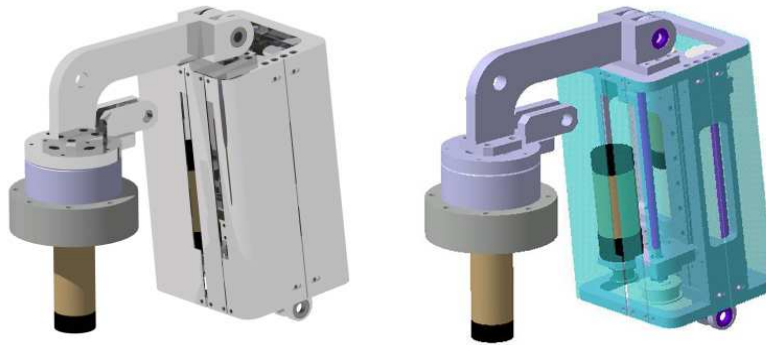


Fig. 3.3 3D model of the shoulder joint actuator

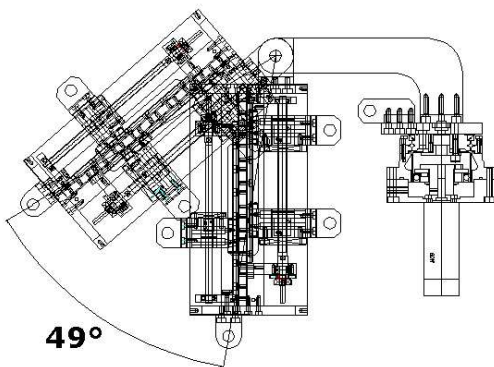


Fig. 3.4 Rotation range of the shoulder pitch joint



Fig. 3.5 Shoulder joint

팔꿈치관절은 Fig. 3.8 과 같은 구조이며 두 번째 pitch축이다. Fig. 3.7 과 같이 몸체의 바깥쪽으로 52°, 안쪽으로 41°의 운동범위를 갖는다.

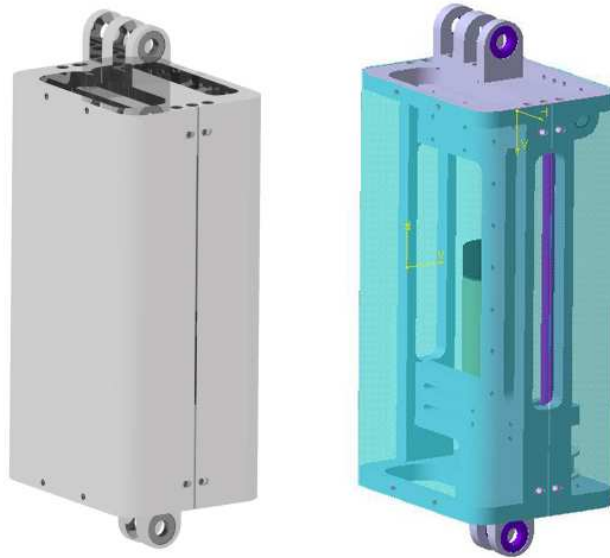


Fig. 3.6 3D model of the elbow joint actuator

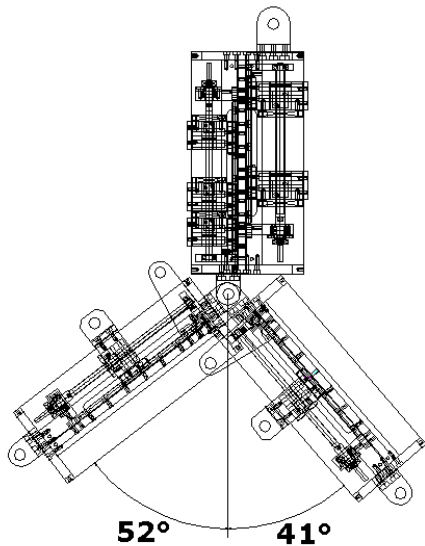


Fig. 3.7 Rotation range of the elbow pitch joint

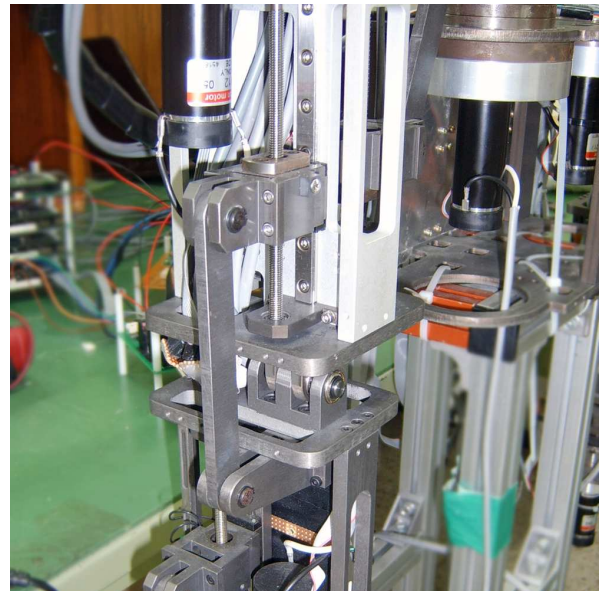


Fig. 3.8 Elbow joint

손목부분은 Fig. 3.9 와 같이 3번째 pitch축과 roll축, 그리고 말단장치 그리퍼 이렇게 3개의 관절로 구성된다. roll축은  $180^{\circ}$ 의 운동범위를 가지며 pitch축은 Fig. 3.10 과 같이 바깥쪽  $53^{\circ}$ , 안쪽  $43^{\circ}$ 의 범위 안에서 운동이 이루어진다.

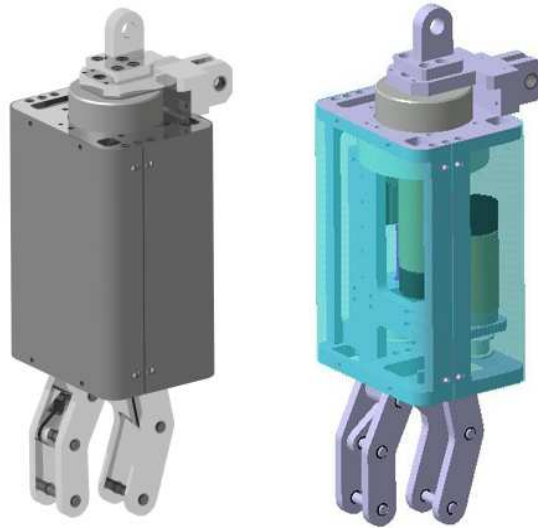


Fig. 3.9 3D model of the wrist joint actuator

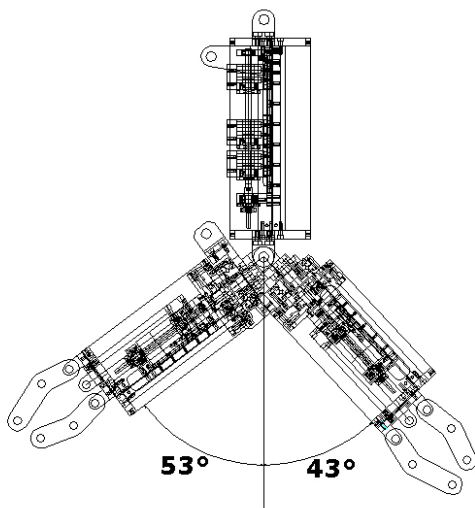


Fig. 3.10 Rotation range of the wrist pitch joint

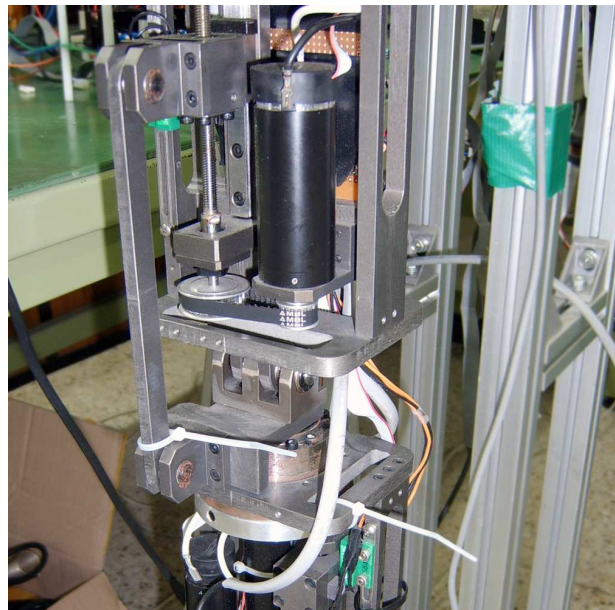


Fig. 3.11 Wrist joint

Table 3.2 는 양팔에 사용된 관절구동기의 사양을 나타낸다.

**Table 3.2 Specification of the joint actuator for arm part**

| 축 | 관절구조 | 운동      | 모터출력<br>[W] | 폴리비   | 감속비     | 볼스크류<br>리드[mm] |
|---|------|---------|-------------|-------|---------|----------------|
| 0 | 직결   | yaw     | 60          | -     | 1 : 160 | -              |
| 1 | 4절링크 | pitch   | 150         | 1 : 1 | 1 : 4.3 | 2              |
| 2 | 4절링크 | pitch   | 60          | 1 : 2 | -       | 1              |
| 3 | 4절링크 | pitch   | 60          | 1 : 2 | -       | 1              |
| 4 | 직결   | roll    | 60          | -     | 1 : 100 | -              |
| 5 | 4절링크 | gripper | 18          | 1 : 2 | -       | 1              |

양팔에 사용된 4절링크 관절을 제어하기 위해, Fig. 3.12 와 같이 네 변중 세 변의 길이는 고정이고 나머지 한 변은 볼스크류의 변위로 만들어 볼스크류의 지지부와 관절각을 고정각으로 하여 관절각과 볼스크류 변위의 관계를 구했다.

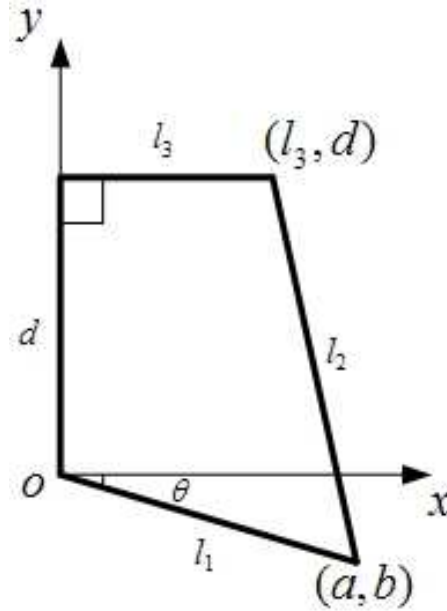


Fig. 3.12 Four-bar link modeling

첫 번째 링크  $l_1$ 에 의해 결정되는 임의의 점을  $(a, b)$ 라 하면

$$a = l_1 \cos \theta, \quad b = l_1 \sin \theta \quad (3.2.1)$$

이고 중심이  $(a, b)$ 이고 반지름이  $l_2$ 인 원의 방정식은 식 (3.2.2) 이다.

$$(x - a)^2 + (y - b)^2 = l_2^2 \quad (3.2.2)$$

점  $(l_3, d)$ 을 위 식에 대입하여  $d$ 에 관하여 정리하면

$$d^2 - 2bd + a^2 + b^2 + l_3^2 - l_2^2 - 2al_3 = 0 \quad (3.2.3)$$



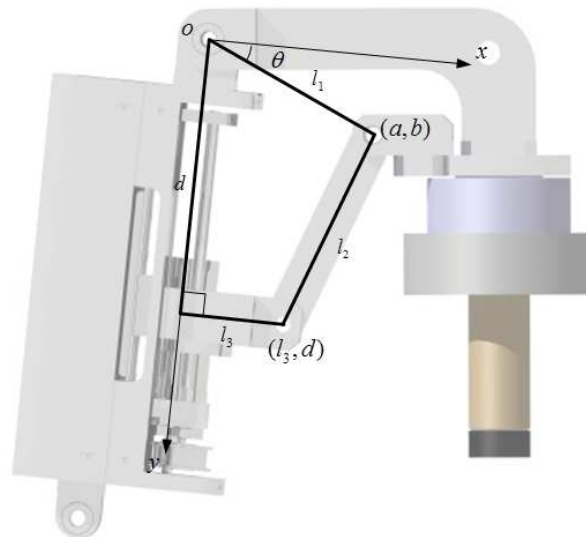
위의 2차방정식을 풀면

$$d = b \pm \sqrt{(l_2 + a - l_3)(l_2 - a + l_3)} \quad (3.2.4)$$

Table 3.3 은 각 4절링크의 변위  $d$ 를 제외한 나머지 링크길이  $l_1$ ,  $l_2$ ,  $l_3$ 를 나타낸 것이다.

**Table 3.3 Length of the link**

| 부분  | $l_1$ [mm] | $l_2$ [mm] | $l_3$ [mm] |
|-----|------------|------------|------------|
| 어깨  | 92.78      | 101.84     | 46.0       |
| 팔꿈치 | 62.68      | 122.0      | 52.0       |
| 손목  | 60.8       | 154.0      | 52.0       |



**Fig. 3.13 Four-bar link modeling for the shoulder**

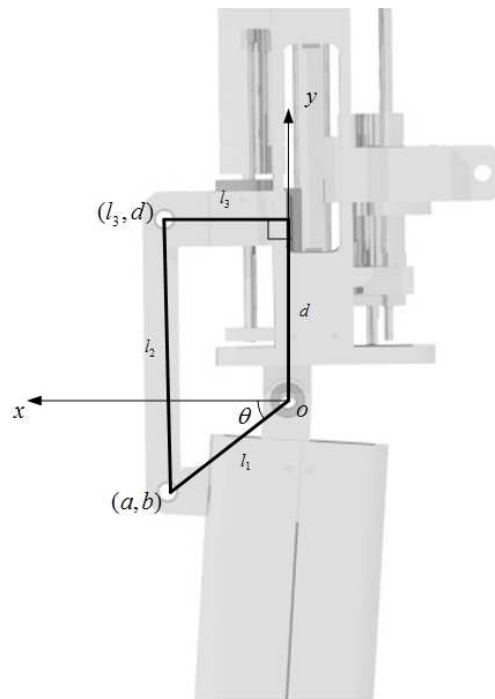


Fig. 3.14 Four-bar link modeling for the elbow

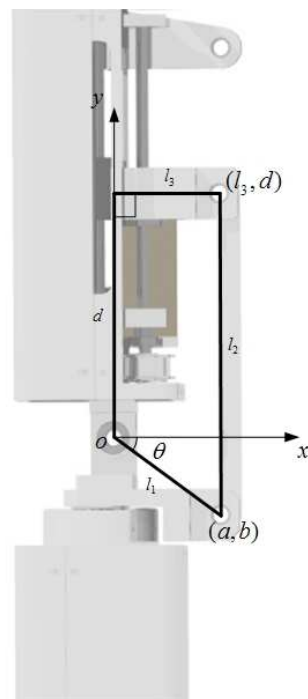


Fig. 3.15 Four-bar link modeling for the wrist



### 3.3 순기구학 해석

로봇은 일련의 강체링크가 관절로 연결된 것이라고 생각할 수 있다. 로봇의 기구학을 해석하기 위해서는 각 링크의 좌표계를 체계적으로 선택할 수 있어야 한다. 로봇 응용에 보편적으로 많이 사용하는 좌표계 선정방법은 Denavit - Hartenberg 규약 또는 D-H 규약이다[6].

좌표계 설정시 Denavit-Hartenberg 규약은 다음과 같다.

(D-H 1)  $x_1$  축은  $z_0$  축과 수직이다.

(D-H 2)  $x_1$  축은  $z_0$  축과 만난다.

D-H 규약은 4개의 기본 변환 행렬의 곱으로 각 관절의 동차 변환 행렬  $A_i$ 로 표현하는 것이다.

$$\begin{aligned}
 A_i &= Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} Rot_{x, \alpha_i} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_ic_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_is_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3.1}
 \end{aligned}$$

여기서  $a_i$ 는 각 관절의 링크길이(length),  $\alpha_i$ 는 비틀림(twist),  $d_i$ 는 오프셋(offset),  $\theta_i$ 는 각도(angle)이다.

로봇 머리의 각 관절의 각과 스테레오비전 시스템에 의해 산출되는 특정물체의 거리에 대한 말단 좌표를 구하기 위하여 머리의 관절구조에 관한 순기구학적 해석을 D-H 규약에 따라 Fig. 3.16 과 같이 좌표계를 설정하였다.

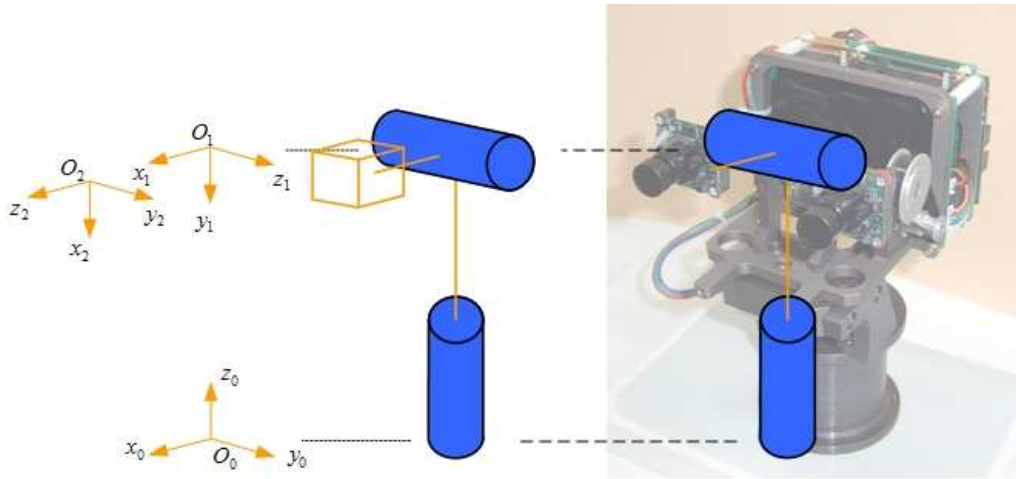


Fig. 3.16 Coordinates of the head

Fig. 3.16 에 의해 설정된 링크 매개변수는 Table 3.4 와 같다.

Table 3.4 Link parameter for head part

| Link | $a_i$   | $\alpha_i$ | $d_i$ | $\theta_i$   |
|------|---------|------------|-------|--------------|
| 1    | 0       | -90        | $d_1$ | $\theta_1^*$ |
| 2    | $a_2^*$ | 90         | 0     | $\theta_2^*$ |

$$(d_1 = 171.5[\text{mm}])$$

$A$  행렬들은 식 (3.31) 과 Table 3.4 의 매개변수로부터 다음과 같이 얻어 진다.

$$A_1 = \begin{bmatrix} c_1 & 1 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} c_2 & 0 & s_2 & a_2 c_2 \\ s_2 & 0 & -c_2 & a_2 s_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.2)$$

따라서 변환행렬  $T$ 는 다음과 같이 주어진다.

$$T_0^2 = A_1 A_2 = \begin{bmatrix} c_1 c_2 + s_2 & s_1 & -c_2 + c_1 s_2 & a_2 (s_2 + c_1 c_2) \\ c_2 s_1 & -c_1 & s_1 s_2 & a_2 c_2 s_1 \\ s_2 & 0 & -c_2 & a_2 s_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.3)$$

그러므로 로봇팔의 말단 위치벡터  $P_{head}$ 는 다음과 같다.

$$P_{head} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} a_2 (s_2 + c_1 c_2) \\ a_2 c_2 s_1 \\ a_2 s_2 + d_1 \end{bmatrix} \quad (3.3.4)$$

로봇 팔 각 관절의 각에 대한 말단 좌표를 구하기 위하여 로봇구조에 관한 순기구학적 해석을 D-H 규약에 따라 Fig. 3.17 과 같이 좌표계를 설정하였다.

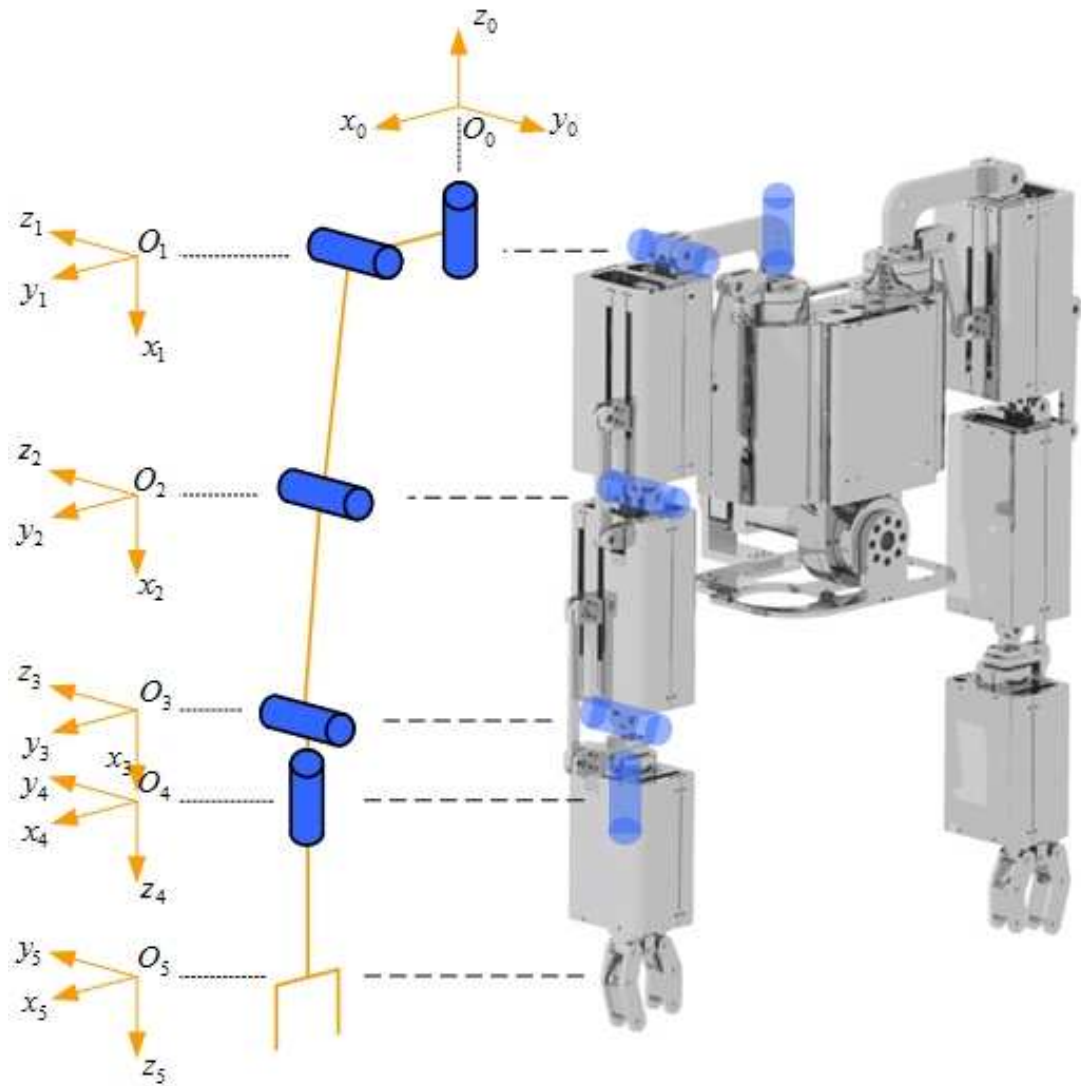


Fig. 3.17 Coordinates of the arm

Fig. 3.17 에 의해 설정된 링크 매개변수는 Table 3.5 와 같다.

Table 3.5 Link parameter for arm part

| Link | $a_i$ | $\alpha_i$ | $d_i$ | $\theta_i$   |
|------|-------|------------|-------|--------------|
| 1    | $a_1$ | 90         | 0     | $\theta_1^*$ |
| 2    | $a_2$ | 0          | 0     | $\theta_2^*$ |
| 3    | $a_3$ | 0          | 0     | $\theta_3^*$ |
| 4    | 0     | 90         | 0     | $\theta_4^*$ |
| 5    | 0     | 0          | $d_5$ | $\theta_5^*$ |

$$(a_1 = 141[\text{mm}], a_2 = 240.35[\text{mm}], a_3 = 231[\text{mm}], d_5 = 235.06[\text{mm}])$$

A 행렬들은 식 (3.3.1) 과 Table 3.4 의 매개변수로부터 다음과 같이 얻어 진다.

$$\begin{aligned}
 A_1 &= R_{z,\theta} \cdot T_{z,d} \cdot T_{x,a} \cdot R_{x,\alpha} = \begin{bmatrix} c_1 & 0 & s_1 & a_1 c_1 \\ s_1 & 0 & -c_1 & a_1 s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_2 &= \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_3 &= \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_4 &= \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_5 &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.3.5}$$

따라서 변환행렬  $T$ 는 다음과 같이 주어진다.

$$T_0^5 = A_1 A_2 A_3 A_4 A_5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3.6)$$

$$\begin{aligned} r_{11} &= c_5 \{c_1 c_4 (c_2 c_3 - s_2 s_3) - c_1 s_4 (s_2 c_3 + c_2 s_3)\} + s_1 s_5 \\ r_{12} &= -s_5 \{c_1 c_4 (c_2 c_3 - s_2 s_3) - c_1 s_4 (s_2 c_3 + c_2 s_3)\} + s_1 c_5 \\ r_{13} &= c_1 s_4 (c_2 c_3 - s_2 s_3) + c_1 c_4 (c_2 c_3 - s_2 s_3) \\ r_{21} &= c_5 \{s_1 c_4 (c_2 c_3 - s_2 s_3) - s_1 s_4 (s_2 c_3 + c_2 s_3)\} - c_1 s_5 \\ r_{22} &= -s_5 \{s_1 c_4 (c_2 c_3 - s_2 s_3) - s_1 s_4 (s_2 c_3 + c_2 s_3)\} - c_1 c_5 \\ r_{23} &= s_1 s_4 (c_2 c_3 - s_2 s_3) + s_1 c_4 (s_2 c_3 + c_2 s_3) \\ r_{31} &= c_5 \{c_4 (s_2 c_3 + c_2 s_3) + s_4 (c_2 c_3 - s_2 s_3)\} \\ r_{32} &= -s_5 \{c_4 (s_2 c_3 + c_2 s_3) + s_4 (c_2 c_3 - s_2 s_3)\} \\ r_{33} &= -c_4 (c_2 c_3 - s_2 s_3) + s_4 (s_2 c_3 + c_2 s_3) \end{aligned}$$

$$\begin{aligned} P_x &= d_5 \{c_1 c_4 (s_2 c_3 + c_2 s_3) + c_1 s_4 (c_2 c_3 - s_2 s_3)\} + c_1 (a_1 + a_2 c_2 + a_3 c_3 c_2 - a_3 s_3 s_2) \\ P_y &= d_5 \{s_1 c_4 (s_2 c_3 + c_2 s_3) + s_1 s_4 (c_2 c_3 - s_2 s_3)\} + s_1 (a_1 + a_2 c_2 + a_3 c_3 c_2 - a_3 s_3 s_2) \\ P_z &= d_5 \{s_4 (c_3 s_2 + c_2 s_3) - c_4 (c_2 c_3 - s_2 s_3)\} + a_2 s_2 + a_3 s_3 c_2 + a_3 c_3 s_2 \end{aligned}$$

그러므로 로봇팔의 말단 위치벡터  $P$ 는 다음과 같다.

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} d_5 \{s_1 c_4 (s_2 c_3 + c_2 s_3) + c_1 s_4 (c_2 c_3 - s_2 s_3)\} + c_1 (a_1 + a_2 c_2 + a_3 c_3 c_2 - a_3 s_3 s_2) \\ d_5 \{s_1 c_4 (s_2 c_3 + c_2 s_3) + s_1 s_4 (c_2 c_3 - s_2 s_3)\} + s_1 (a_1 + a_2 c_2 + a_3 c_3 c_2 - a_3 s_3 s_2) \\ d_5 \{s_4 (c_3 s_2 + c_2 s_3) - c_4 (c_2 c_3 - s_2 s_3)\} + a_2 s_2 + a_3 s_3 c_2 + a_3 c_3 s_2 \end{bmatrix} \quad (3.3.7)$$

### 3.4 역기구학 해석

로봇 매니퓰레이터(manipulator)를 제어하는데 있어 말단장치의 위치와 방향에 대한 관절변수를 역으로 찾아내는 역기구학 문제를 풀 때, 짧은 시간에 방정식을 풀기 위해 반복적인 계산을 하는 수치해보다 해석적 해를 구하는 것이 바람직하다[6].

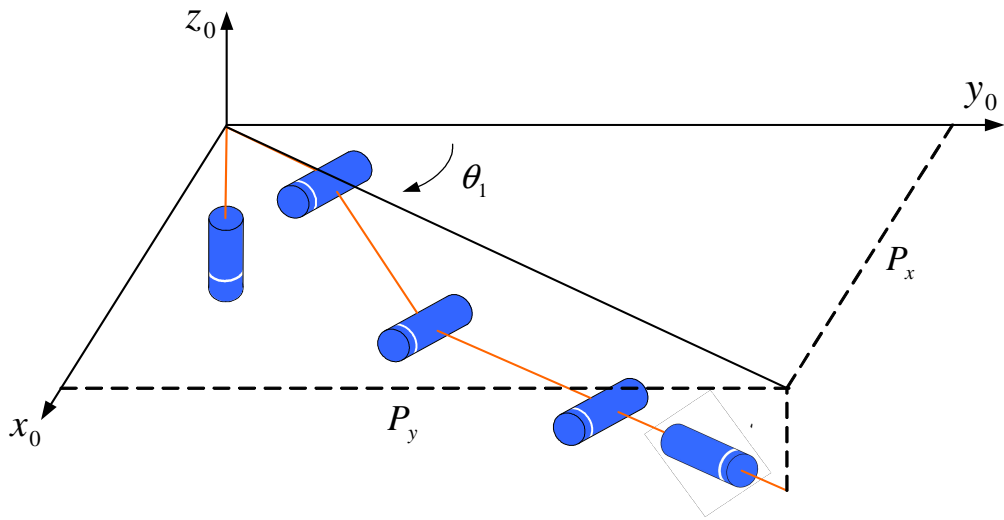


Fig. 3.18 Projection to the  $x_0y_0$  plane

Fig. 3.18 은  $x_0y_0$  평면에 투영한 모습이다. 이 투영에서

$$\theta_1 = \text{Atan}(P_x, P_y) \quad (3.4.1)$$

를 알 수 있고 여기서  $\text{Atan}(x, y)$ 는 이변수 역탄젠트(two argument arctangent) 함수를 나타낸다.

2 ~ 4번째 링크의 운동이 평면적이므로  $\theta_2, \theta_3, \theta_4$ 를 구하기 위해 링크 2,3,4로 형성된 평면을 생각하면 Fig. 3.19 와 같다.

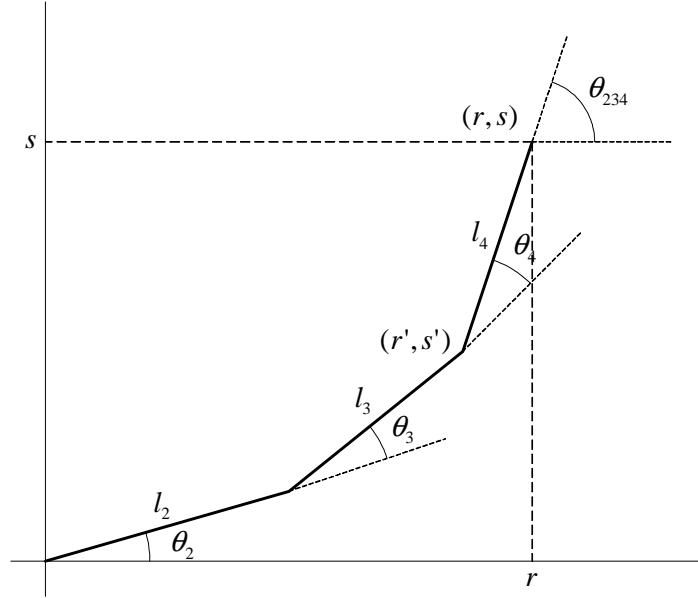


Fig. 3.19 Projection to the plane formed with link 2,3,4

$$\theta_{\phi} = \theta_{234} = \theta_2 + \theta_3 + \theta_4 \quad (3.4.2)$$

라 두면 끝점에서 링크 4를 제외한 임의의 점  $(r, s)$ 는

$$r' = r - l_4 c_{\phi} = l_2 c_2 + l_3 c_{2+3}, \quad s' = s - l_4 s_{\phi} = l_2 s_2 + l_3 s_{2+3} \quad (3.4.3)$$

이고 피타고라스의 정리에 의해

$$r'^2 + s'^2 = l_2^2 + l_3^2 + 2l_2 l_3 c_3 \quad (3.4.4)$$

$c_3$ 와  $s_3$ 에 관해 정리하면



$$c_3 = \frac{r'^2 + s'^2 - l_2^2 - l_3^2}{2l_2l_3}, \quad s_3 = \pm \sqrt{1 - c_3^2} \quad (3.4.5)$$

$$\theta_3 = \text{Atan}(s_3, c_3) \quad (3.4.6)$$

링크 2의 말단좌표를  $(K_1, K_2)$ 라 하면

$$K_1 = l_2 + l_3c_3, \quad K_2 = l_3s_3 \quad (3.4.7)$$

$$\theta_2 = \text{Atan}(s', r') - \text{Atan}(K_2, K_1) \quad (3.4.8)$$

$$\theta_4 = \theta_\phi - (\theta_2 + \theta_3) \quad (3.4.9)$$

여기서  $\theta_\phi$ 는 말단링크의 방향으로서 팔의 모양에 따라 수정되어야 할 변수이며  $\theta_5$ 는 그리퍼의 형상을 결정하는 회전관절이므로 잡을 물체의 형상에 따라 결정해야 할 변수이다.

## 제 4 장 관절제어기의 구성

### 4.1 관절제어기의 하드웨어적 구성

이족보행로봇의 팔을 제어하기 위해 개발된 관절제어기의 구성은 Fig. 4.1 과 같다. 제어기 1개로 관절 2축을 제어할 수 있으며 각각의 제어기는 CAN(Controller Area Network)을 통해 연결되어 있어 상위제어기에서 보내는 제어명령을 수신하여 각각의 관절구동기(joint actuator)인 모터를 제어하고 제어 완료시 완료신호를 송신하게 된다.

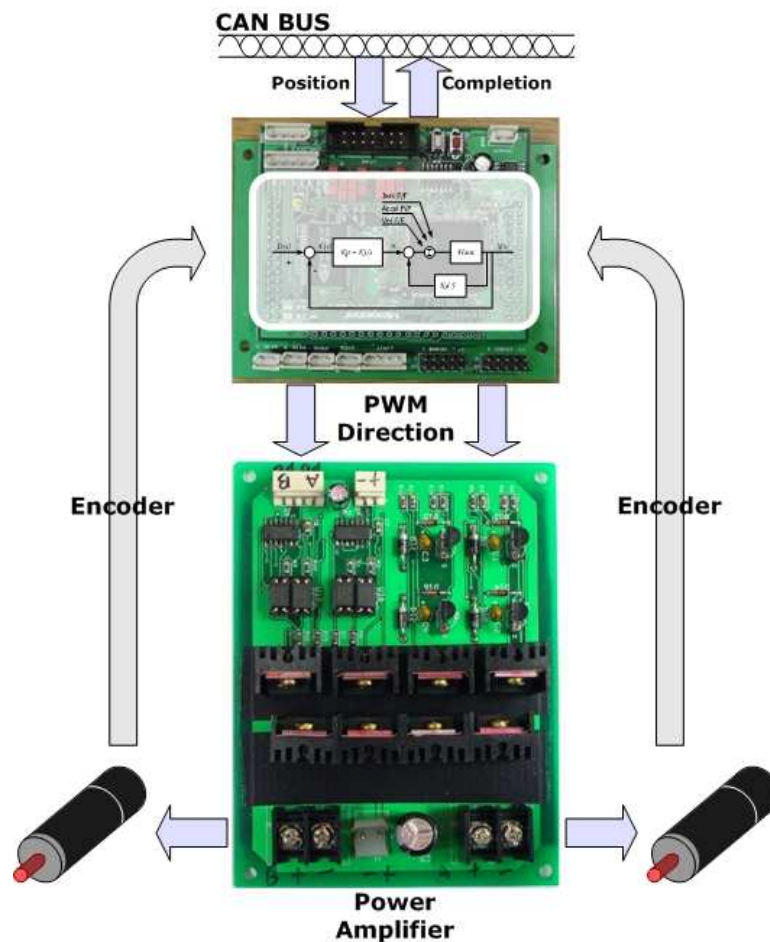


Fig. 4.1 Joint controller

모터컨트롤러는 CAN Bus로부터 제어명령을 수신하게 되면 컨트롤러에 내장되어 있는 PID 제어알고리즘에 의한 PWM(Pulse Width Modulation)파형과 모터 회전방향신호를 전원증폭회로인 모터드라이버로 보내게 되고 드라이버에서는 모터의 정격전압레벨로 증폭시켜주게 되어 모터로 전원을 공급, 모터를 구동하게 된다. 제어가 완료되면 다시 CAN Bus를 통해 제어완료신호를 보내어 상위제어기로 하여금 제어가 완료되었음을 알리게 된다.

로봇의 머리와 팔을 제어하기 위하여 머리 1개와 팔부분에 6개의 관절제어기가 사용되었으며 상위제어기와 CAN통신을 이용하여 네트워크를 구성하여 로봇의 모션을 만들게 된다.

## 4.2 모터제어 알고리즘

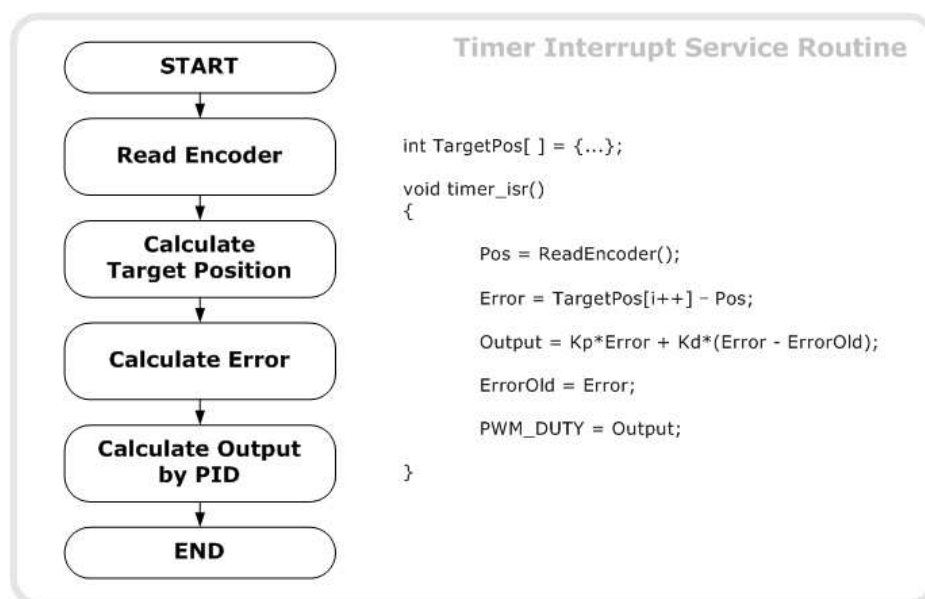


Fig. 4.2 Flow chart of the timer ISR

PID 제어는 단순하고, 높은 효율성 때문에 흔히 사용되는 제어기법이다. Fig. 4.2 는 모터컨트롤러의 타이머인터럽트 서비스루틴(timer interrupt service

routine)에 구현된 PID 제어알고리즘의 실행순서이다. 먼저 모터의 현재 엔코더(encoder)값을 읽고 목표위치 값과의 오차를 계산, PID 제어식에 의해 모터출력을 결정하여 PWM 듀티비(duty)에 적용하게 된다. 이로 인해 모터는 원하는 회전위치로의 제어가 가능하게 된다.

Fig. 4.3 은 라플라스 함수를 이용한 PID 제어기의 블록선도이다.

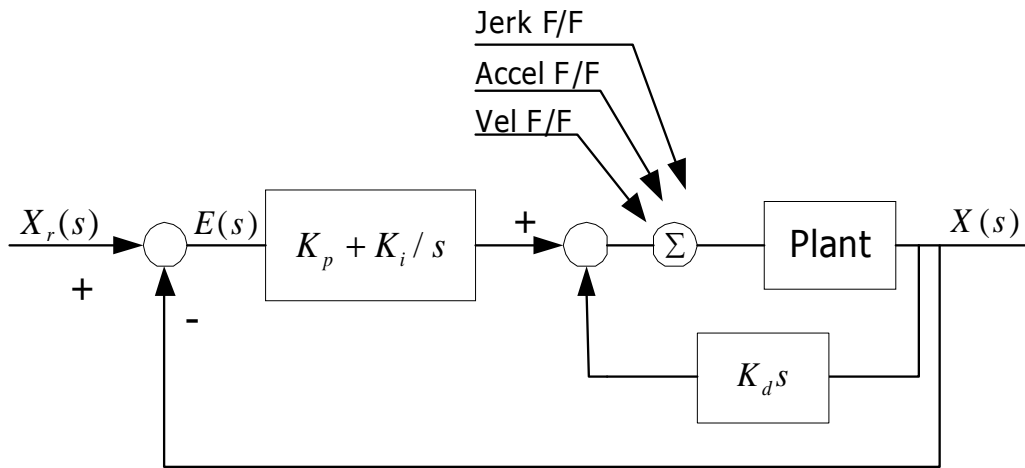


Fig. 4.3 Block diagram of the PID controller

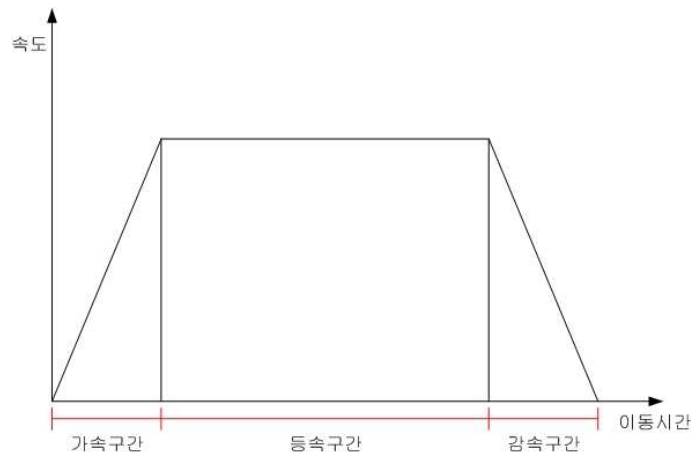


Fig. 4.4 Velocity profile

모터의 위치제어가 가능하게 되면 Fig. 4.4 와 같은 사다리꼴 형태의 속도제어가 이뤄지는데 이는 모터가 기동시 발생하는 관성을 줄여 모터를 보호하고 로봇

관절의 갑작스런 동작으로 인한 기구부의 파손을 방지하기 위함이다. 속도제어는 모터가 회전하는 전체이동량에서 가감속구간을 두어 그 부분에서는 이동량을 천천히 변화시키는 방법으로 이뤄지게 된다.

### 4.3 CAN 통신을 이용한 제어기의 네트워크 구성

CAN(Controller Area Network)은 초기에 자동차산업분야에 적용하기 위해 고안된 시리얼 네트워크 통신방식이며 근래에는 자동차분야뿐만 아니라 산업 전 분야에 폭넓게 적용되고 있다. 임베디드 시스템(embedded system)에서 일반적으로 CAN 버스는 마이크로 프로세서(micro-processor) 사이에서 통신망을 형성하며, 2가닥의 꼬임선(twist pair wire)으로 연결되어 반이중통신(half duplex) 방식으로 짧은 메시지를 사용하는 고속응용시스템에 적당하다. 또한 외부요인인 노이즈(noise) 등에 강인성을 가지고 있어 통신 에러율을 최소화하여 높은 신뢰성을 보인다.

이러한 장점으로 CAN은 본 연구에서 개발된 이족보행로봇 제어시스템에서 사용되어진 분산제어방식에 매우 적합한 통신방식이다.

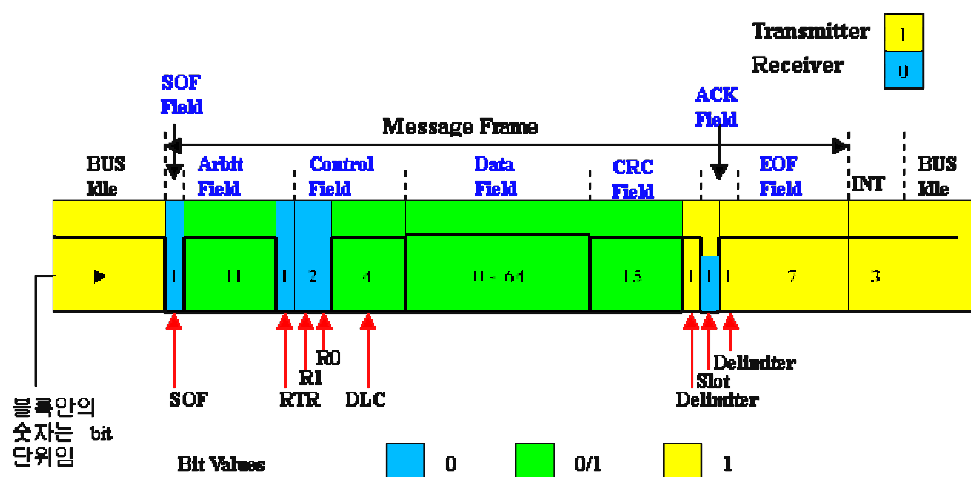


Fig. 4.5 Message frame of CAN 2.0a

Fig. 4.5 는 로봇에 사용된 CAN 2.0A의 메시지프레임 구조이다. 11개의 비트(bit)로 구성된 식별자영역과 64비트의 데이터영역으로 구성되며 식별자영역에 수신측의 고유식별자(ID)가 표시되어 데이터를 해당 프로세서로 전송하게 된다 [8].

개발된 로봇에 사용된 모터제어용 프로세서는 7개로써 Table 4.1 과 같이 프로세서 및 모터 고유식별자(ID)를 설정하였다. ID는 ‘[A][B][C]’와 같이 세자리 정수로 표현되며 ‘[A]’ 자리는 프로세서가 속한 로봇의 기구적 부분을 나타내고 ‘[B]’ 자리는 그 부분 안에서 프로세서 고유순서, ‘[C]’ 자리는 프로세서인지 모터인지를 나타내게 된다.

**Table 4.1 Formed CAN ID**

| 자리  | 값   | 내용      | 비고                            |
|-----|-----|---------|-------------------------------|
| [A] | 0   | 주제어기    | 주제어기 : 010                    |
|     | 1   | 다리부분    |                               |
|     | 2   | 팔부분     |                               |
|     | 3   | 머리부분    |                               |
| [B] | 0~9 | 프로세서 번호 | 다리 : 0~6<br>팔 : 0~5<br>머리 : 0 |
| [C] | 0   | 프로세서    |                               |
|     | 1   | 모터 A    |                               |
|     | 2   | 모터 B    |                               |

CAN 통신에서 1회 송신시 가능한 데이터양은 64비트이며 일반적으로 16비트씩 나누어 Mail Box A, B, C, D라 칭한다. 통신 내용은 Table 4.2 와 같이 6종

류의 신호로 구성하였는데, 관절의 구동전원을 스위칭(switching)할 수 있는 Amp ON/OFF신호와 모터의 제어치, 그리고 관절제어기에서 주제어기로 보내지는 제어완료신호와 에러신호가 있다.

**Table 4.2 Contents of CAN**

| 통신방향               | 내용          | MailBox A<br>(16 bit) | MailBox B<br>(16 bit)        | MailBox C<br>(16 bit) | MailBox D<br>(16 bit) |
|--------------------|-------------|-----------------------|------------------------------|-----------------------|-----------------------|
| 주제어기<br>↓<br>관절제어기 | Amp ON      | 100                   | 0                            | 0                     | 0                     |
|                    | Amp OFF     | 200                   | 0                            | 0                     | 0                     |
|                    | 모터 A<br>제어치 | 300                   | 원점 : 0<br>정방향 : 1<br>역방향 : 2 | 제어치의 상위<br>16비트       | 제어치의 하위<br>16비트       |
|                    | 모터 B<br>제어치 | 400                   |                              |                       |                       |
| 관절제어기<br>↓<br>주제어기 | 제어완료<br>신호  | 500                   | 모터<br>식별자                    | 0                     | 0                     |
|                    | 에러신호        | 600                   | 프로세서<br>식별자                  | 0                     | 0                     |

## 제 5 장 Stereo Vision System의 구성

통상적으로 로봇이 전면에 위치한 물체를 감지하고 그 위치를 인식하기 위하여 초음파, 적외선, 레이저, 비전 등 여러 가지 종류의 센서가 사용되어진다. 본 연구에서는 그 중에서 인식율도 좋고 인식조건의 변경이 용이한 스테레오 비전 시스템(stereo vision system)을 사용하였다.

일반적으로 비전시스템은 CCD 카메라(charge-coupled device camera)와 카메라로부터 들어오는 아날로그신호를 A/D변환기를 통해 디지털화하여 메모리에 저장하는 이미지 프레임그래버(image frame grabber)가 사용되는데, 2대의 CCD 카메라를 사용하는 스테레오비전에서는 이미지그래버에 의한 두 이미지간의 동기가 중요하다. 이미지그래버에서는 카메라로 트리거(trigger) 신호를 보내어 각각의 이미지데이터가 획득되는 시간을 맞추게 된다.

획득된 이미지데이터를 처리하기 위하여 공개 영상처리 라이브러리인 OpenCV(Open source Computer Vision library)[9]를 사용하였으며 이미지그래버의 메모리로부터 데이터를 가져오기 위하여 Matrox사에서 제공하는 MIL-Lite 라이브러리를 사용하였다.

각각의 이미지를 영상처리하여 특정물체의 각 2D좌표를 구한 후 물체와 두 카메라사이간의 거리를 산출하였으며 카메라가 장착되어 있는 로봇 머리부분의 구조에 따른 순기구학식을 이용, 물체가 위치하고 있는 3D 좌표를 계산하였다.

### 5.1 Vision System의 구성 및 사양

개발된 스테레오 비전시스템의 구성 및 사양은 아래 Fig. 5.1 , Table 5.1 과 같다.



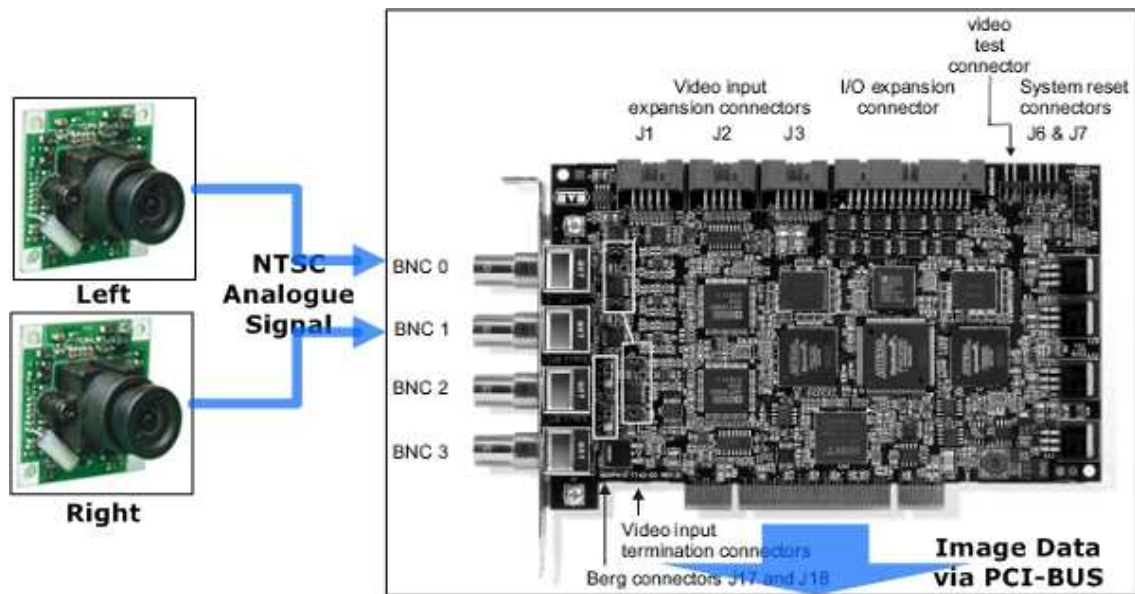


Fig. 5.1 Stereo vision system

Table 5.1 Specification of the stereo vision system

| 구분                   | 사양             | 비고   |
|----------------------|----------------|--|
| CCD Camera           | tcc-3232H      | 1/3" CCD Image Sensor,<br>Electronic Shutter |
| Image Grabber        | Matrox Morphis | 4ch NTSC input,<br>PCI interface             |
| 영상획득용<br>S/W Library | MIL-Lite       |  |
| 영상처리용<br>S/W Library | OpenCV         | Open Source<br>Computer Vision Library       |
| Frame Rate           | 약 25 ~ 30 fps  | 2 Camera, no processing                      |

개발에 사용된 Matrox Morphis 이미지그래버는 동시에 두 개의 표준 아날로그

신호를 획득할 수 있으며, 자체적으로 16MB의 저장공간을 가지고 있고 부가적 기능으로 실시간 'JPEG' 포맷으로의 압축이 가능하다. Fig. 5.2 는 Mophis의 내부구조를 나타낸 것이다.

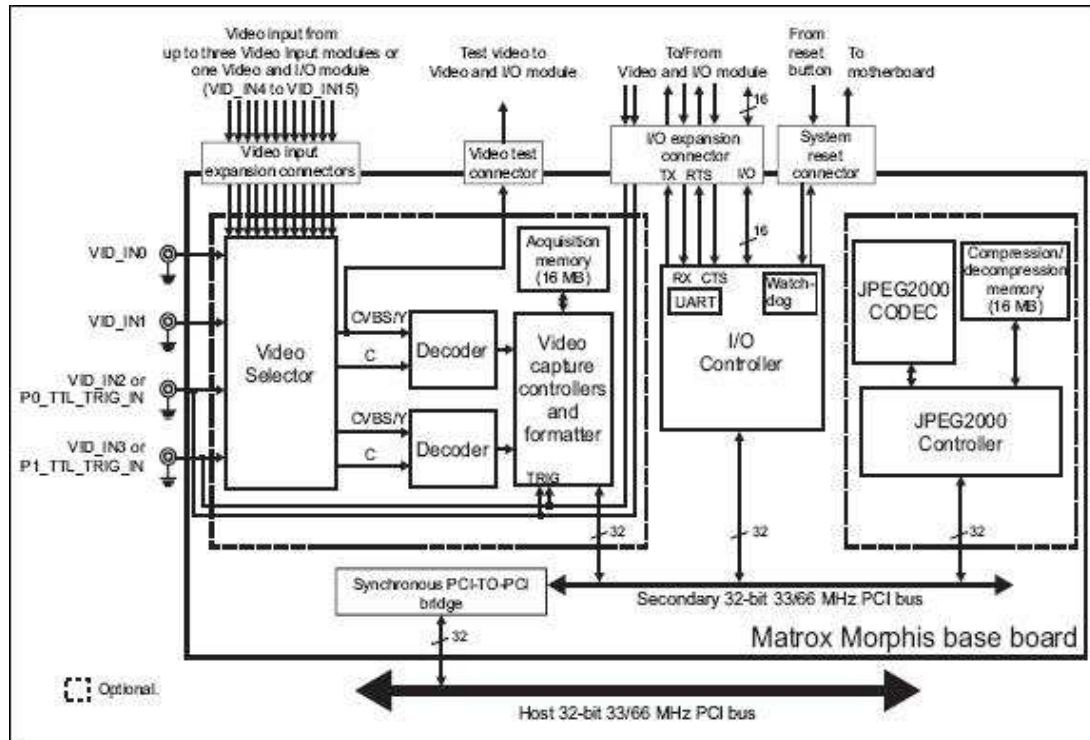


Fig. 5.2 Internal structure of the image grabber 'Morphis'

## 5.2 OpenCV를 이용한 영상처리 및 물체거리 산출

본 연구에서는 스테레오비전시스템의 영상처리를 위해 OpenCV 라이브러리를 사용하였다. OpenCV 라이브러리(Open source Computer Vision library)는 인텔(Intel) 사의 주도 하에 만들어지는 영상처리 전용 라이브러리이다. OpenCV에서 제공하는 알고리즘은 영상읽기, 출력 등 간단한 기능에서부터 고급알고리즘까지 폭넓게 제공되며 다양한 알고리즘 외에도 본 연구에서 주제어기로 사용된 인텔 계열의 CPU에 최적화되어 있기 때문에 빠른 실행속도를 보여준다는 장점이 있다[8].

스테레오비전을 이용한 특정물체의 거리산출은 Fig. 5.3 과 같은 형태로 이루어진다.

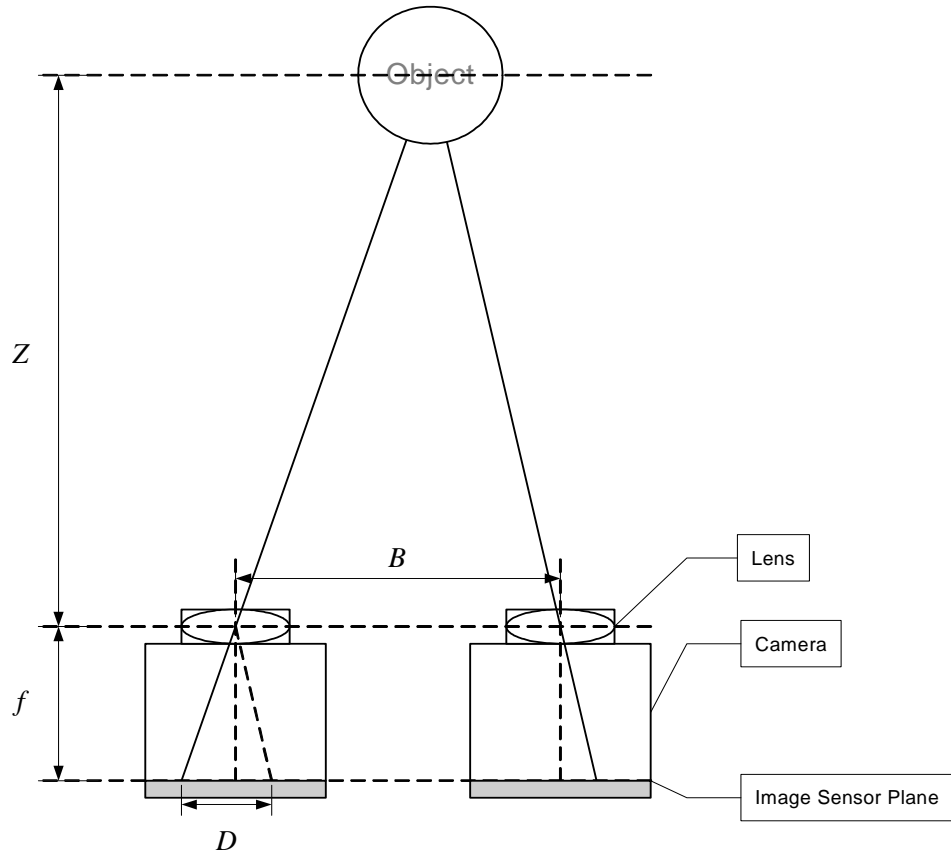


Fig. 5.3 Theory for computing the distance using the stereo vision system

물체거리  $Z$  는 식 (5.3.1) 의 비례식과 같이 이미지센서 상의 영상 거리차  $D$  와 두 카메라의 거리  $B$  의 관계와 렌즈에서 이미지센서까지의 거리  $f$  에 의하여 구할 수 있다.

$$f : Z = D : B, \quad Z = \frac{fB}{D} \quad (5.3.1)$$

여기서  $D$  는 이미지센서상의 물체의 영상이 맺히는 위치의 차이로써 식

(5.3.2) 와 같이 각 카메라로부터 입력받은 영상에 대하여 영상처리를 통해 검출된 두 이미지간의  $x$ 좌표 차  $d$  에 의해 구할 수 있다.

$$D = \frac{dW}{w}, \quad d = x_2 - x_1 \quad (5.3.2)$$

따라서, 물체거리  $Z$  는 식 (5.3.3) 에 의해 구할 수 있게 되는 것이다.

$$\therefore Z = \frac{fBw}{W(x_2 - x_1)} \quad (5.3.3)$$

$$(B = 70[\text{mm}], \quad W = 6[\text{mm}], \quad w = 320, \quad f \approx 5.5[\text{mm}])$$

또한 산출된 물체거리  $Z$  로부터 3장에서 언급한 로봇의 머리에 대한 순기구학 식 (3.3.4) 를 이용하면 물체의 실제 3D 좌표를 구할 수 있다.

## 제 6 장 x86기반의 로봇 주제어기

대부분의 이동하는 로봇은 독립주행을 위해서 임베디드 시스템(embedded system)을 구축하고 있다. 그렇지 못한 로봇은 이동거리 및 동작에 제한을 받게 된다. 로봇이 독립적으로 동작하기 위해서 가장 중요한 것은 전원, 통신 및 구동기이다. 이 모든 것을 관장하는 이족보행로봇의 주제어기는 로봇의 센서 및 관절 제어기 등으로부터 들어오는 모든 정보를 획득할 수 있어야 하며, 주어진 상황에 대한 자체적인 판단, 그리고 외부로 로봇상태에 관한 정보를 알릴 수 있어야 한다. 또한 로봇에 탑재하기 위해 크기도 작아야 하며, 소비전력은 낮을수록 좋다. 본 연구에서는 이러한 조건을 충족하기 위해 소형 임베디드 컴퓨터를 사용하였다.

임베디드 컴퓨터에는 이미지그래버가 설치되어 있어 눈에 장착된 CCD카메라로부터 들어오는 영상정보를 획득 및 처리할 수 있으며 내장된 시리얼통신포트와 CAN통신모듈을 이용하여 관절구동기나 센서로부터 여러 가지 정보를 획득할 수 있다. 또한 무선랜카드를 사용하여 외부로 로봇의 현재정보를 알릴 수 있고 반대로 제어명령을 수신하여 명령에 대한 동작도 가능하다.

임베디드 컴퓨터의 운영체제(operating system)로는 일반적으로 Desktop PC에 사용하는 'Windows XP'를 설치하여 Win32환경에서 로봇 구동프로그램이 개발되었다. 개발된 프로그램은 앞에서 언급한 주제어기의 여러 가지 기능을 구현함과 동시에 특히 로봇제어기에서 중요시 되는 실시간성(real-time)을 최대한 보장하기 위하여 Multi-thread 방식으로 개발되었다.

### 6.1 Win32 Multi-thread/Message 기반의 프로그램

운영체제에서 프로세스(process)란 현재 메모리에 적재되어 실행중인 프로그램을 말한다. 프로세스는 실행중인 프로그램이지만 실제로 작업을 수행하는 주체는

아니다. 작업은 프로세스내의 스레드(thread)가 담당하게 되는데 하나의 프로세스가 생성되면 최소한 한 개 이상의 스레드가 생성된다. 프로세스와 동시에 만들어지는 스레드를 주 스레드(primary thread)라 하는데, Multi-Thread라 함은 주 스레드 이외에 스레드를 추가로 생성하여 사용하는 것을 일컫는다.

멀티스레드 프로그래밍의 가장 큰 장점은 Fig. 6.1 에서 볼 수 있듯이 작업의 병렬화이며 이로 인해 빠른 처리속도와 CPU사용률 증가 등의 효과를 기대할 수 있다. 또한 명확하게 정의되어진 독립된 작업을 전담하는 개별적인 스레드를 사용함으로써 프로그램의 설계를 단순화 할 수 있다[10].

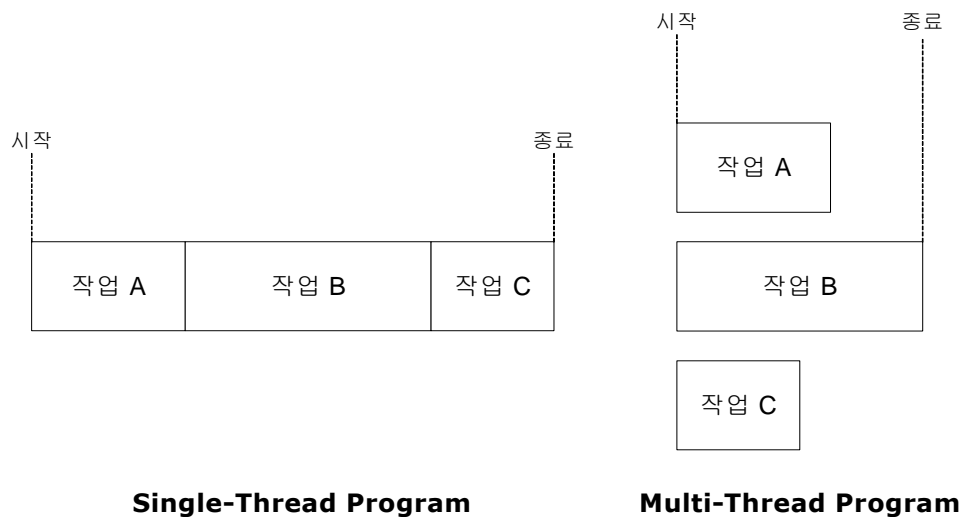


Fig. 6.1 General idea for the multi-threaded programming

또한 운영체제에서 제공하는 스레드 스케줄러(thread scheduler)에 의해 스레드들이 관리되는데 이미 검증되어진 운영체제가 스레드를 관리함으로써 프로그램 전체의 안정성도 향상된다.

독립적으로 구동되는 스레드를 서로 연결하여 정보를 교환하기 위한 스레드간 통신방법으로는 공유메모리, 익명파이프, 메시지 등의 방법이 있는데, 본 연구에서는 구조가 간단하고 사용이 용이한 메시지(message)를 사용하였다.

Windows는 메시지기반의 운영체제로써, 응용프로그램은 시스템이 보내는 미리

정해진 메시지를 기다리고 메시지를 받게 되면 해당하는 동작을 구동하는 이벤트드리븐(event-driven)방식의 형태로 동작을 수행하게 된다. 미리 정해진 메시지 외에도 개발자가 메시지를 정의하여 사용할 수 있는데, 스레드는 이런 메시지들을 수신할 수 있는 일종의 버퍼인 메시지 큐(message que)를 가지고 있으며 수신할 스레드의 핸들(handle)을 알면 스레드로 메시지를 보낼 수 있다. 스레드에서는 큐에 새로 들어온 메시지가 있는지 검사하고, 만약 있다면 메시지내용에 해당하는 작업을 수행하게 된다. 이로써 프로세스내의 스레드간 통신이 이루어져 독립적인 작업들을 관리할 수 있게 되는 것이다. Table 6.1 은 Win32 Message의 구조를 나타낸 것이다[12].

**Table 6.1 Windows Message**

| 구성      | 변수형           | 크기     | 내용        |
|---------|---------------|--------|-----------|
| Message | unsigned int  | 32 bit | 메시지       |
| WPARAM  | unsigned int  | 32 bit | 메시지의 부가정보 |
| LPARAM  | unsigned long | 32 bit | 메시지의 부가정보 |

## 6.2 프로그램의 구성

개발된 휴머노이드 로봇의 주제어기는 각 관절에 대한 제어량을 계산해야 되고 내장된 스테레오비전 시스템을 위해 영상처리와 물체거리를 산출해야하며 각 관절제어기와 통신이 가능해야 한다. 또한 외부조종기로부터 들어오는 명령을 인식하고 수행해야하며 자체적으로 내부에 구성된 여러 제어모듈을 관리감독하고 오류발생시 조치할 수 있어야한다.

이와 같은 조건을 충족하기 위해 앞에서 언급한 Win32 Multi-Thread 기반으로 프로그램을 작성하였으며 각 모듈간의 통신을 위하여 윈도우 메시지를 이용

하였다. 프로그램의 전체구성은 Fig. 6.2 와 같다.

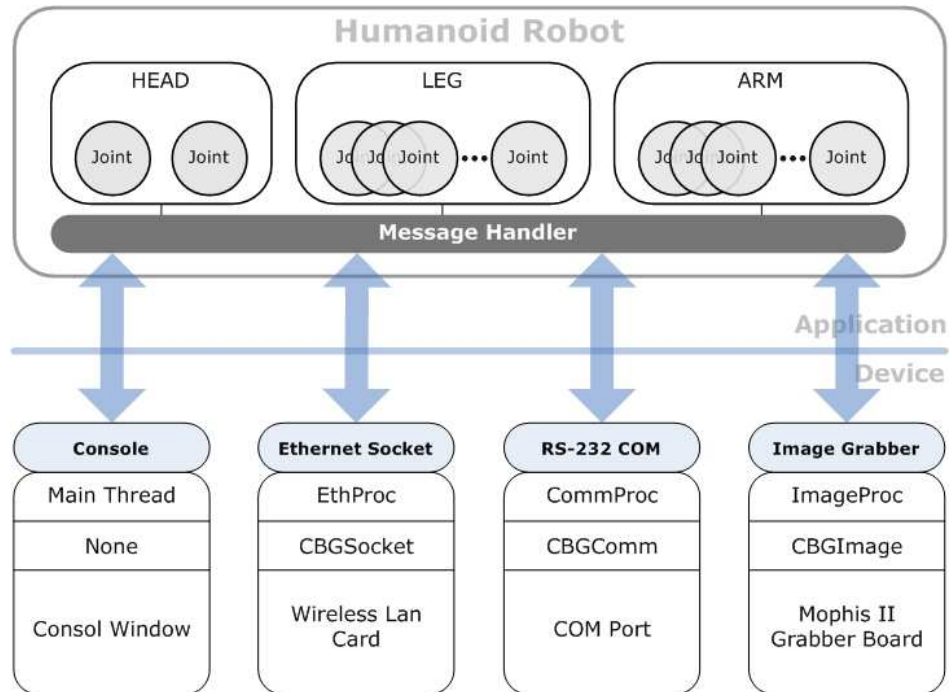


Fig. 6.2 Block diagram of the main controller program

프로그램은 크게 실질적인 로봇제어에 관한 응용계층과 제어기에 설치된 여러 장치를 관장하는 장치계층으로 나뉜다.

응용계층에서는 각 관절의 제어량을 계산하고 로봇의 내부와 외부에서 일어나는 모든 상황에 대처하기 위해 휴머노이드 로봇을 추상화하여 클래스로 구현하였다. 클래스 내부에는 메시지 핸들러(message handler) 함수를 정의하여 장치계층으로부터 들어오는 메시지를 처리할 수 있도록 하였다.

장치계층에서는 기본 입출력을 위한 콘솔윈도우와 외부와의 통신을 위한 이더넷소켓, 관절제어기와 통신을 위한 직렬통신포트, 그리고 영상처리를 위한 이미지그래버에 대한 기능을 정의하였다. 각 장치모듈은 각각 무한루프의 스레드함수를 두어 장치를 실시간으로 감시할 수 있도록 하였다.



다음은 각 장치모듈에 대한 설명이다.

(1) 콘솔윈도우(console window) : 주로 개발과정에서 사용되었으며 가장 기본적인 입출력 장치로써 기본적인 로봇동작을 명령할 수 있고 또한 현재의 로봇의 상태를 확인할 수 있다.

(2) 이더넷소켓(ethernet socket) : 외부 조종기와의 무선랜통신을 관장하는 모듈이다. 조종기에서 들어오는 명령내용을 응용계층으로 전달하며 응용계층의 데이터를 외부에서 모니터링(monitring)할 수 있도록 데이터를 전송한다.

(3) 직렬통신포트(RS-232c port) : 관절제어기 네트워크에 접속되어 각 관절제어기와 통신하는 역할을 한다. 관절의 제어량을 전송하며 반대로 관절제어기로부터 들어오는 동작완료신호와 에러신호를 수신하고 응용계층에 알리게 된다.

(4) 이미지그래버(image grabber) : 로봇의 머리에 장착된 두 대의 카메라로부터 영상데이터를 처리하여 산출된 물체의 거리를 응용계층으로 전달한다.

## 제 7 장 3D 시뮬레이션

실제 로봇의 구동실험에 따르는 시간적 손실과 실험간 발생하는 위험성을 줄이기 위하여 가상의 3차원공간에 실제와 같은 조건의 로봇 모델을 두고 물리력을 적용할 수 있는 3D 그래픽 시뮬레이터를 개발하였다.

Fig. 7.1 과 같은 구성으로 개발된 시뮬레이터는 윈도우즈환경에서 동작하고 로봇의 관절을 임의로 제어할 수 있으며 또한 목표하는 물체의 좌표를 설정, 역기구학을 이용한 팔제어를 통해 목표물체에 도달할 수 있는 기능을 포함하고 있다.

개발을 위해 Microsoft Visual C++ .NET 컴파일러가 사용되었고[11], 3D 그래픽환경을 위하여 Silicon Graphics 사의 OpenGL(Open source Graphics Library)[13]과 물리력을 적용하기 위하여 공개물리엔진인 ODE(Open source Dynamics Engine) 라이브러리[14]가 사용되었다.

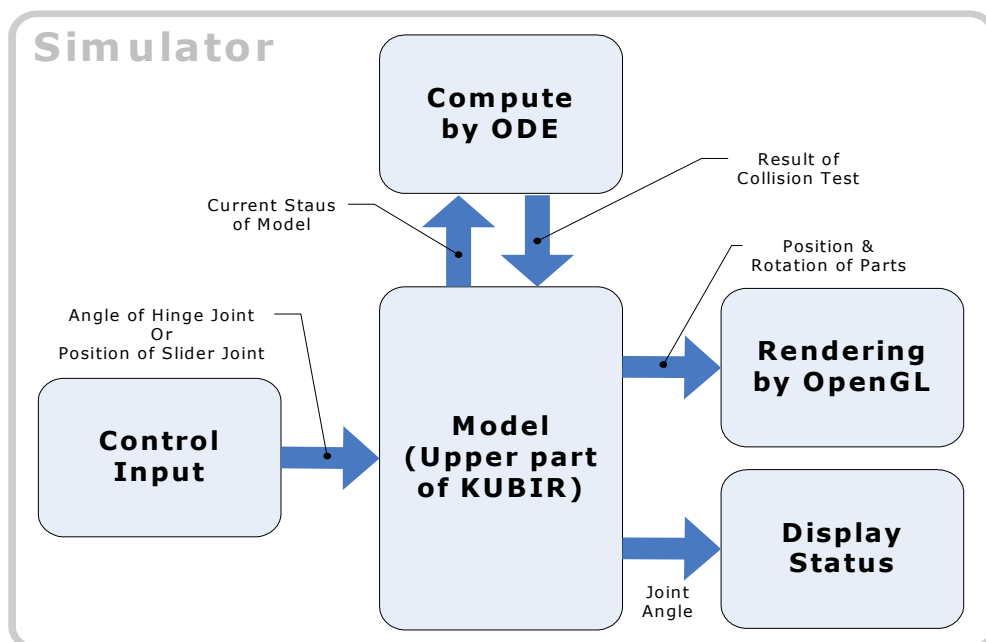


Fig. 7.1 Block diagram of the 3D simulator

## 7.1 시뮬레이터의 구성

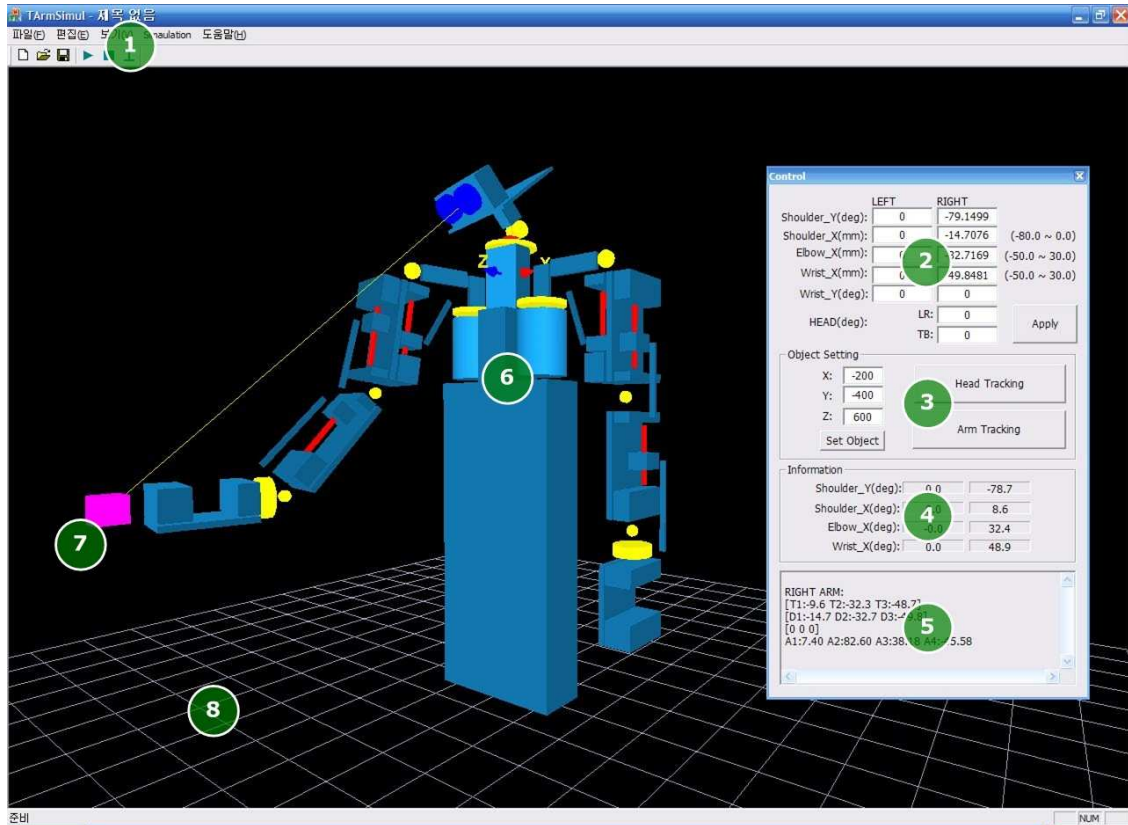


Fig. 7.2 3D graphical simulator

3D 시뮬레이터의 화면구성은 Fig. 7.2 와 같으며 그 기능은 다음과 같다.

- ① 메뉴 및 툴바 : 시뮬레이션을 시작하거나 종료할 수 있으며 초기화할 수 있다. 시뮬레이션이 시작되면 로봇에 중력 등의 물리력이 적용되며 각 부재간의 충돌검사가 이뤄진다.
- ② 관절제어입력부 : 로봇의 관절을 임의로 제어할 수 있는 부분으로서 해당하는 관절부분에 제어값을 입력하여 관절을 제어할 수 있다.
- ③ 물체위치설정부 : 물체의 좌표를 설정할 수 있으며 그에 따라 머리와 팔을 역

기구학에 의해 물체를 트래킹할 수 있게 되어있다.

④ 관절각표시부 : 현재 제어되고 있는 각 관절의 각을 표시한다.

⑤ 로그표시부 : 계산과정 등 기타 여러 가지 정보를 표시한다.

⑥ 로봇모델 : 강체동역학을 시뮬레이션 하기 위해 ODE엔진을 사용하여 구성되었으며 시뮬레이션 실행 시 물리력이 적용되어 진다.

⑦ 목표물체 : 임의의 좌표로 설정된 물체로써 머리 및 팔의 트래킹의 목표물이 된다.

⑧ 가상공간지면 : 가상공간상의 지면으로서 모델링은 지면을 기준으로 하여 이뤄진다.

## 7.2 물리엔진의 소개

개발된 시뮬레이터는 실제와 유사한 물리 법칙을 필요로 한다. 물리학 관점에서 로봇은 여러 개의 조인트를 연결한 강체의 집합이다. 하나의 부재에 적용되는 힘과 토크는 연결된 부재에 영향을 미치고 반발력이 작용할 수 있다. 따라서 모든 부재는 중력에 영향을 받게 되므로 가상의 강체 동역학을 시뮬레이션할 툴이 필요하다. 강체를 시뮬레이션 할 수 있는 라이브러리는 많이 있지만 가격은 고가이다. 그래서 무료이면서 검증된 ODE(Open source Dynamics Engine)이라는 물리 엔진을 사용하였다.

ODE 물리 엔진에서 사용되는 부재는 질량 중심에 해당하는 위치 벡터, 선속도, 부재의 방향, 각속도 벡터와 같은 성질을 가진다. 이와 같은 성질들은 시간에 따

라 변하는 동적인 것이며 정적인 것은 부재의 질량, 질량의 중심 위치와 관성 행렬이다. 또한 Fig. 7.3 과 같이 볼(ball)과 소켓(socket), 힌지(hinge), 유니버설(universal), 슬라이더(sliding) 등의 다양한 조인트를 사용할 수 있다.

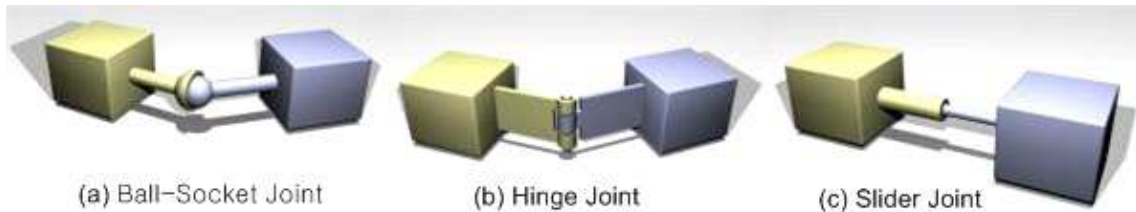


Fig. 7.3 Joint types of ODE

개발된 로봇은 팔부분 12자유도와 머리부분 2자유도, 총 14자유도를 가지므로 14개의 힌지 조인트와 사절링크로 구성된 어깨와 팔꿈치, 손목의 관절을 구동하기 위한 6개의 슬라이더 조인트로 이루어져 있고 그 외에 부재를 연결하기 위한 고정과 힌지 조인트로 구성된다.

### 7.3 로봇 모델링

시뮬레이션을 위해서는 실제로봇의 크기와 구조, 질량 등의 물성치를 추상화하여 3D 가상공간에 모델링하여야 하는데 이를 위해서 각각의 부재를 박스 형태와 실린더 형태의 오브젝트를 만들어 실제로봇의 2D 및 3D 도면을 참조하여 최대한 실제와 흡사하도록 모델링하였다.

각각의 부재를 표현하기 위해 박스오브젝트 70개와 실린더오브젝트 22개가 사용되었으며 부재를 접합하기 위해 고정조인트를, 관절의 움직임을 위해 힌지 및 슬라이더 조인트를 이용하여 로봇을 구성하였다.

모델링된 로봇은 ODE 물리엔진에 의해 각 부재의 충돌 및 구속여부가 계산되

고 그 결과, 각 부재의 위치 및 회전성분을 구할 수 있게 되며 이것을 참조하여 OpenGL 라이브러리를 이용, 화면에 그려지게 된다.

## 7.4 관절제어시스템 구성

실제 로봇의 팔은 모터를 이용하여 그 자세가 제어되는데, 만약 팔을 든 상태에서 구동전원을 차단하면 팔은 중력의 작용으로 그 자세를 유지할 수 없게 된다. 마찬가지로 시뮬레이터의 가상공간에서도 중력이 작용하므로 각각의 조인트는 실제 모터와 같은 위치제어를 필요로 하게 된다. 이를 위해서 실제 모터제어에 사용되는 PID 제어 알고리즘을 이용하여 각 관절제어시스템을 구성하였다.

PID 제어는 자동제어 방식 가운데서 가장 흔히 사용되는 제어방식으로 비례( $K_p$ ), 적분( $K_i$ ), 미분( $K_d$ )의 3 가지 게인(gain)을 조합으로 다양하고 유연한 제어가 가능해진다. 이는 오차신호  $e(t)$ 를 피드백(feedback) 받아서 제어신호  $u(t)$ 를 계산한다.

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (7.4.1)$$

원래 PID 제어는 식 (7.4.1) 과 같이 연속한 아날로그량을 제어하는 것이 기본으로 되어 있다. 그러나 시뮬레이터의 데이터 입출력은 일정시간 간격을 가지고 미/적분 연산의 빠른 데이터 처리를 위해 고성능의 컴퓨터가 필요하게 되므로 식 (7.4.2) 와 같이 이산화를 해야 한다.

$$u(n) = u(n-1) + \Delta u(n)$$

$$\Delta u(n) = K_p [e(n) - e(n-1)] + K_i e(n) + K_d [(e(n) - e(n-1)) - (e(n-1) - e(n-2))] \quad (7.4.2)$$

시뮬레이터에서는 부재의 지지를 위해 각 관절의 모터부에 토크 출력 값이 필요하므로 관절의 모터부를 제어하기 위해서 식 (7.4.3) 을 이용해 다음과 같은 토크 제어 신호  $\tau$  를 얻을 수 있다.

$$\tau(n) = \tau(n-1) + \Delta\tau(n)$$

$$\Delta\tau(n) = K_p[\theta(n) - \theta(n-1)] + K_i\theta(n) + K_d[(\theta(n) - \theta(n-1)) - (\theta(n-1) - \theta(n-2))] \quad (7.4.3)$$

## 제 8 장 실험 및 고찰

실험은 특정물체를 카메라 앞에 두고 획득되는 영상에서 물체를 검출하여 이미지의 물체중심좌표를 가지고 머리의 트래킹모션, 거리검출, 팔제어의 순서로 진행하였다. 머리와 팔의 구동은 안정적이었고 이론적인 제어동작을 구현할 수 있었으나 2개의 이미지를 처리하는 시간이 비교적 길어 처리시간동안에 제어 명령이 없어 동작이 부자연스럽다는 문제점을 확인할 수 있었다. 영상처리로 인한 지연시간동안 동작을 제어할 수 있도록 옵저버(observer)기반 비주얼서보잉의 필요성을 인식할 수 있었다[7]. 거리검출실험에서는 가까운 거리에서는 스테레오 비전 시스템의 선형성을 확인할 수 있었으나 사용된 카메라의 넓은 화각으로 인해 물체가 카메라로부터 멀어질수록 정확성이 감소함을 확인할 수 있었다.

### 8.1 물체검출 및 거리측정 실험



Fig. 8.1 Image processing of the vision system

Fig. 8.1 은 두 개의 영상이미지에서 사각형을 검출하여 물체의 중심좌표를 구



하는 과정이다. 이를 통하여 Fig. 8.2 와 같이 머리의 트래킹동작실험을 하였다.

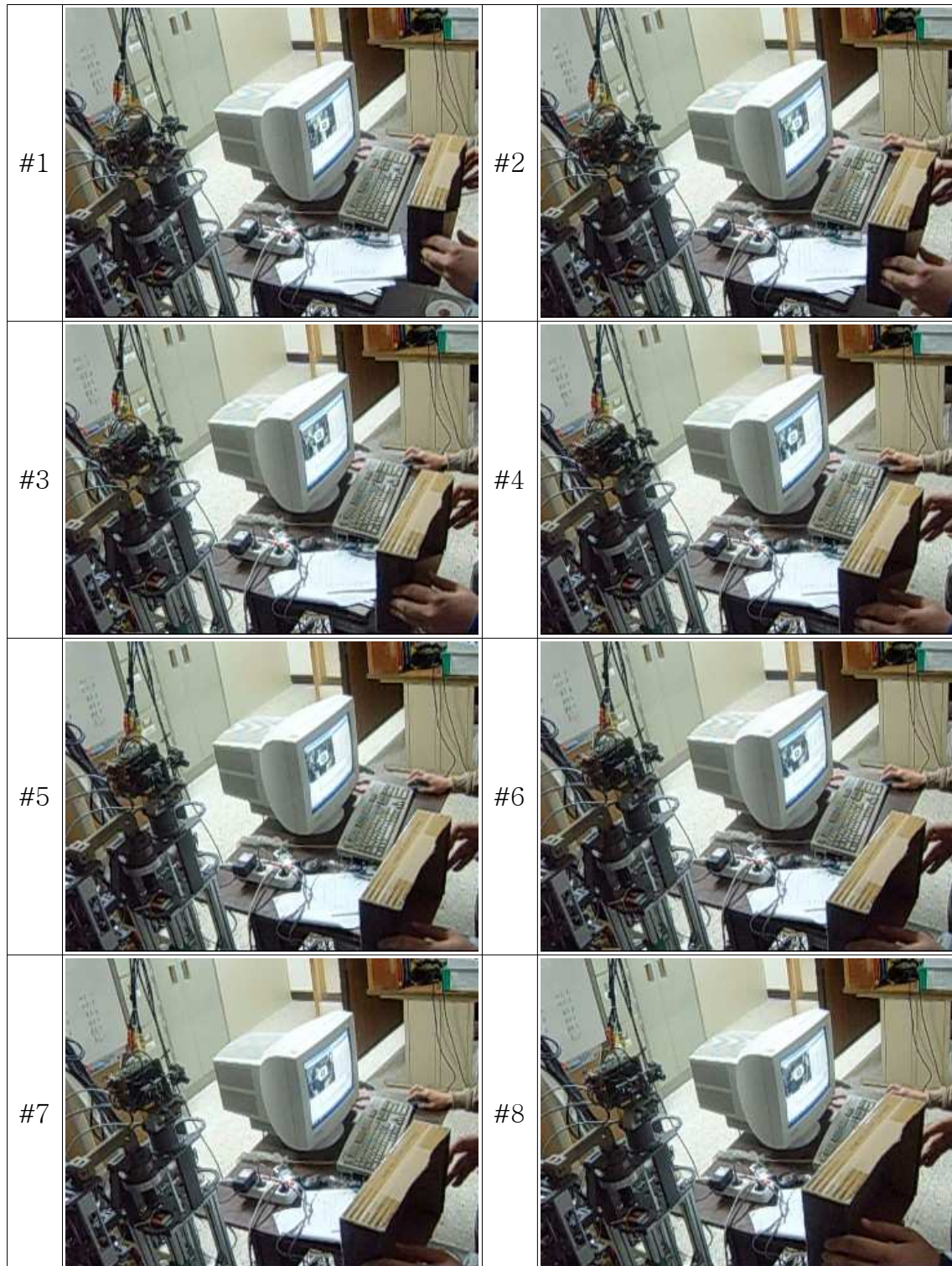


Fig. 8.2 Tracking experiment of the head

Fig. 8.3은 트래킹동작시의 이미지영상을 나타낸 것이다.

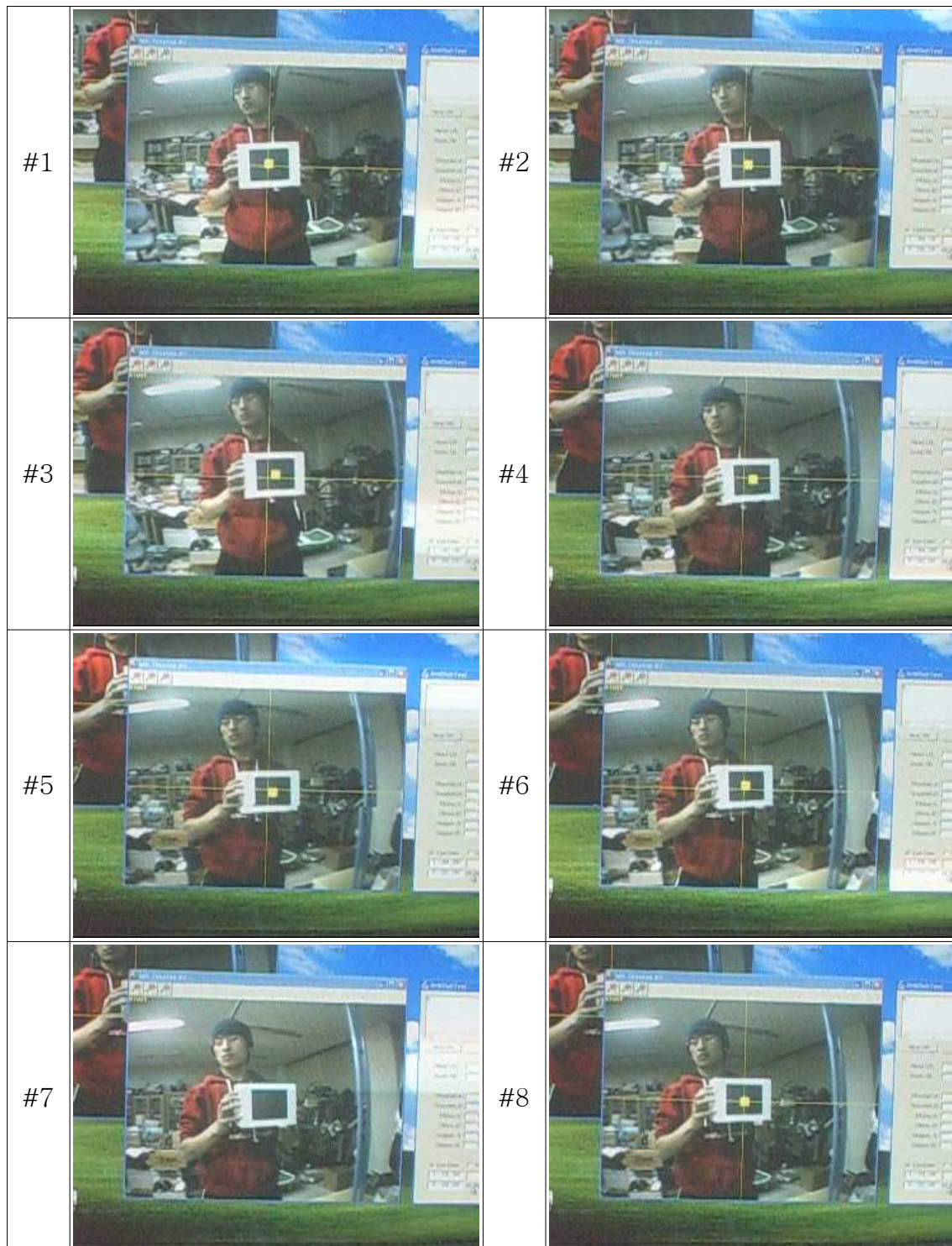


Fig. 8.3 View of tracking control



두 개의 영상으로부터 영상처리를 통해 물체를 검출하고 5장의 식 (5.3.3)을 이용하여 물체거리를 검출하였으며 그 결과는 Fig. 8.4 와 같다.

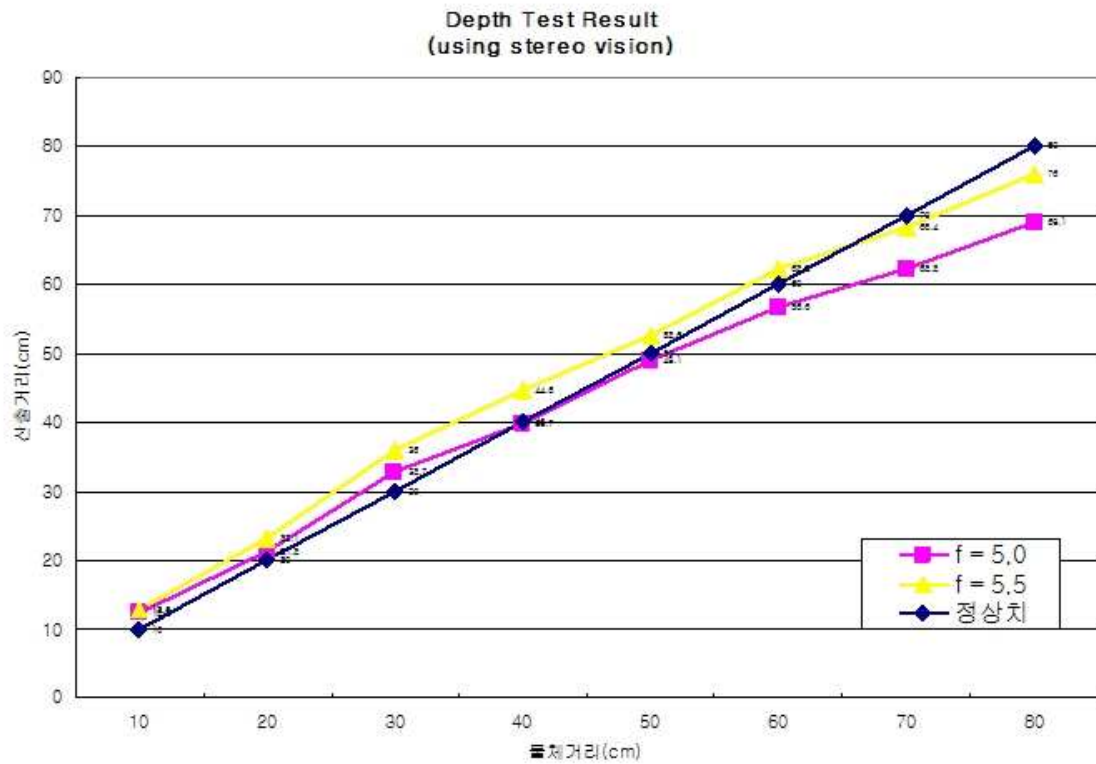


Fig. 8.4 Result of computing distance using the stereo vision system

## 8.2 팔의 모션제어

스테레오비전 시스템에 의해 산출된 물체거리와 물체를 바라보고 있는 머리의 관절각을 가지고 Fig. 8.5 과 같이 3장에서 언급한 머리부의 말단 위치벡터  $P_{head}$ 를 이용하여 물체의 좌표를 구했다. 3.4절의 말단좌표에 따른 역기구학의 닫힌 해를 이용하여 각 관절의 각을 구했으며 팔을 제어하여 물체에 그리퍼가 도달할 수 있었다.

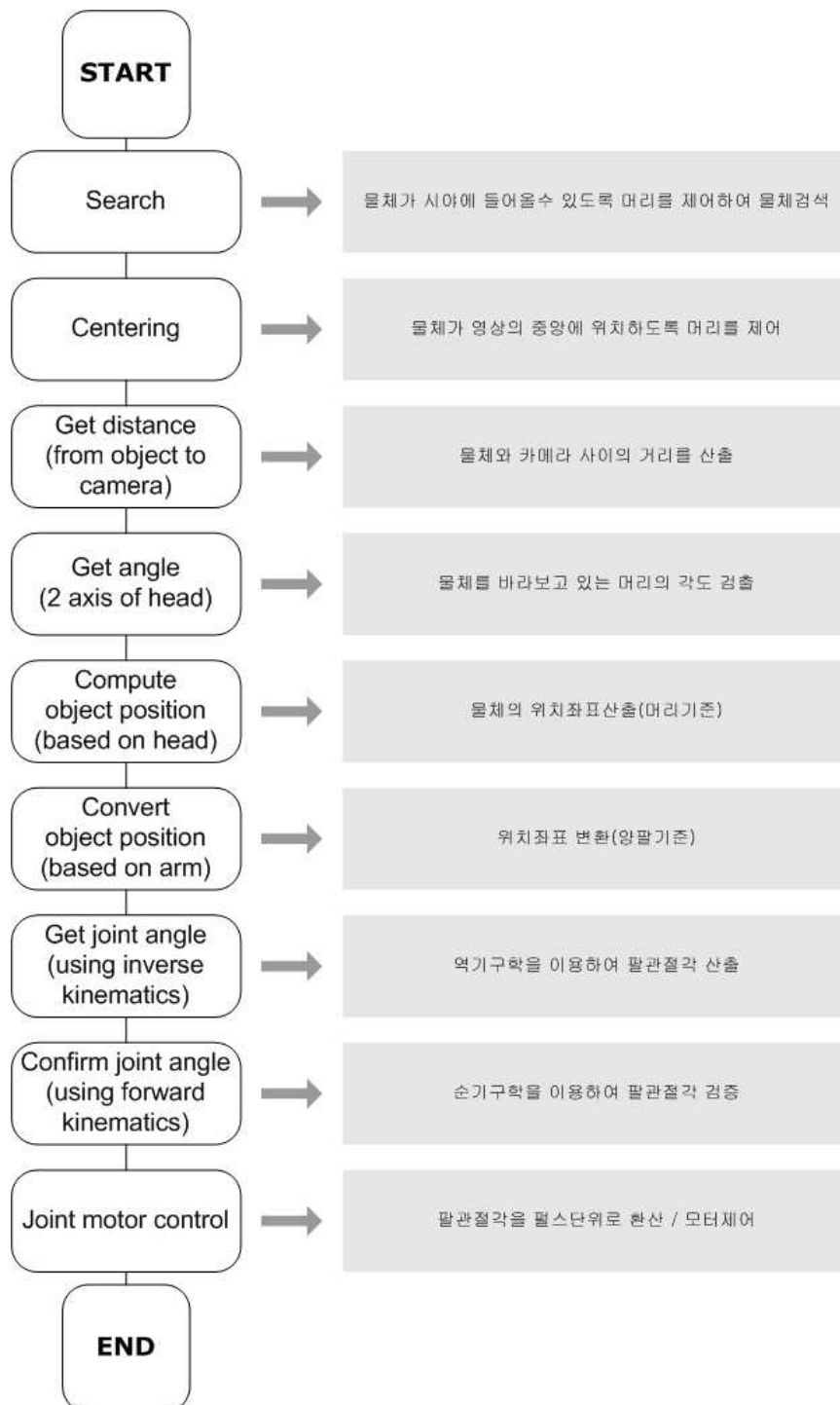


Fig. 8.5 Flow chart of visual servoing

Fig. 8.5 는 팔의 모션을 제어하는 그림이다.

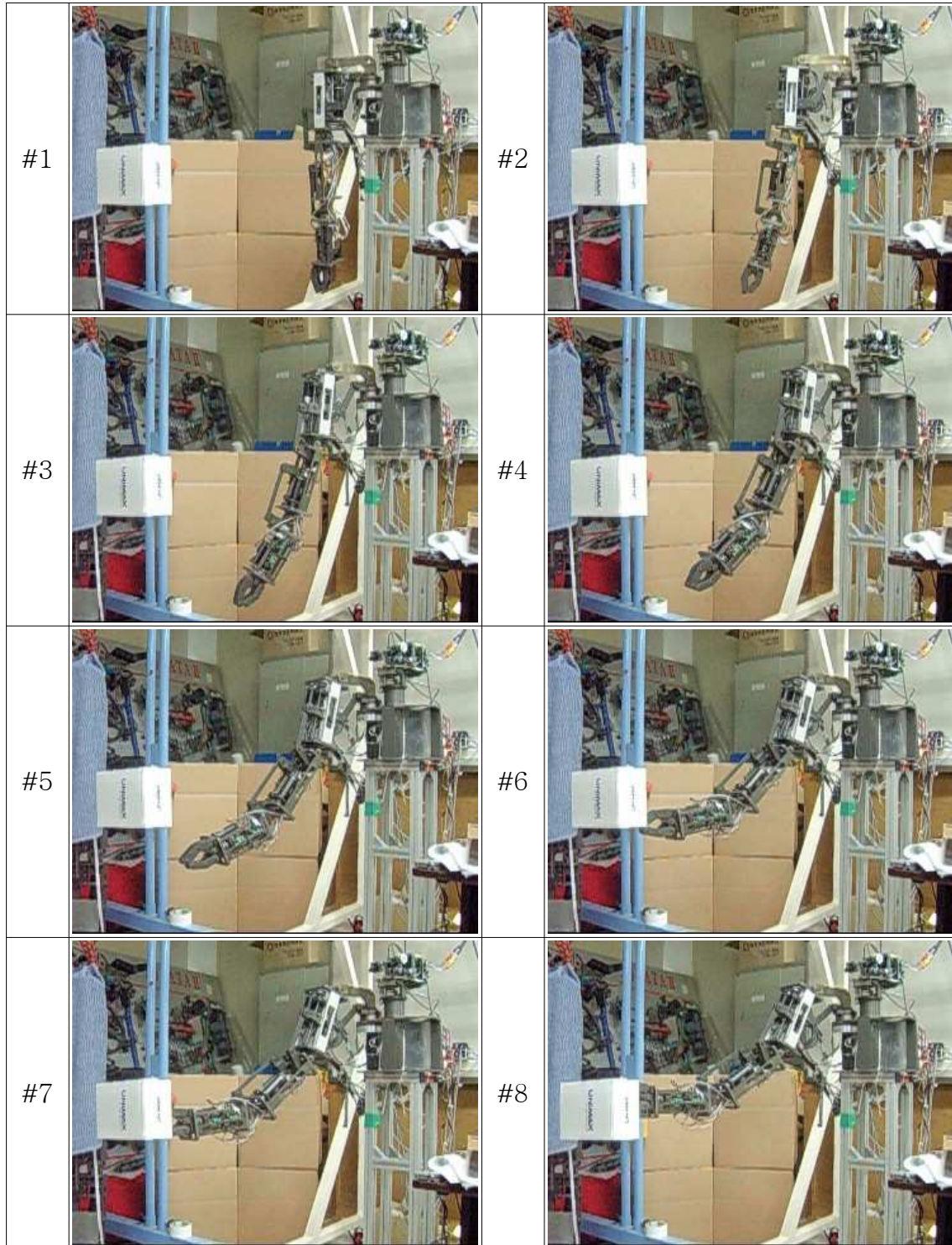


Fig. 8.6 Motion of the arm

## 제 9 장 결 론

본 논문에서는 양팔을 갖는 이족보행로봇 'KUBIR(Korea Maritime University Biped Robot)'에 2대의 CCD 카메라와 이미지그래버를 이용하여 스테레오비전 시스템을 구축하였으며 카메라를 이용한 비주얼서보잉 제어시스템에 대하여 연구하였다.

본 연구의 대상인 상체 12자유도의 양팔에 대하여, 말단장치좌표의 제어를 위한 팔의 관절각을 변화시킬 수 있는 역기구학의 닫힌 해(closed-form)를 구했으며, 비전시스템을 이용하여 산출된 특정물체의 위치를 말단장치좌표와 매핑(mapping)하여 비주얼서보잉을 구현하였다.

비주얼서보잉 제어를 적용한 양팔로봇의 구동실험에 앞서 3D 그래픽 시뮬레이터를 개발하여 연구된 제어이론의 타당성을 검증하였으며 실제 팔 구동실험을 통해 검증된 이론이 제어시스템에 잘 적용되었음을 확인하였다.

향후 휴머노이드 로봇의 제어시스템 성능개선을 위해서는, 주제어기의 실시간성을 향상시키기 위한 리얼타임 운영체제(real-time operation system)나 RTX 등과 같은 실시간 라이브러리의 사용이 필수적이며, 비전시스템에 있어서는 보다 고속의 영상처리방법을 통한 물체인식율의 향상이 필요하다.

## 참 고 문 헌

- [1] A.Takanishi, Y. Egusa, M. Tochizawa, M. Takeya, and I. Kato, "Realization of Dynamic Biped Walking Stabilized with Trunk Motion" RoManSy 7: Proc. Seventh CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators, A.Morecki, G.Bianchi and KoJaworek, Eds.Hermes, Paris, pp.68~79, 1990.
- [2] <http://asimo.honda.com>
- [3] J.H Kim, S.W Park, I.W Park, and J. H. Oh, "Development of a Humanoid Biped Walking Robot Platform KHR-1 - Initial Design and Its Performance Evaluation", in Proceedings of 3rd IARP Int. Workshop on Humanoid and Human Friendly Robotics, pp.14~21, Tsukuba, Japan, Dec. 2003.
- [4] Bradley J.Nelson, Pradeep K. Khosla, "An Extendable Framework for Expectation-Based Visual Servoing Using Environment Models", IEEE trans. on Robotics and Automation, Vol. 1, pp. 184~189, 1995.
- [5] Shahram Jafari, Ray Jarvis, "Relative Visual Servoing: A Comparison with Hybrid Intelligent Models", 2nd International Conference on Autonomous Robots and Agents December 13-15, 2004.
- [6] Spong, Vidyasagar, "Robot Dynamics and Control", WILEY.
- [7] Edited by B. K. Ghosh, Ning Xi, T. J. Tarn, "Control in Robotics and Automation Sensor-Based Intergration", Academic Press
- [8] TMS320LF2407A User Guide, Texas Instrument. Inc.
- [9] Open Source Computer Vision Library - Reference Manual, Intel

- [10] Aaron Michael Cohen , Mike Woodring, "Win32 멀티스레드 프로그래밍", 한빛미디어
- [11] 이상엽, "Visual C++ Programming Bible", 영진출판사
- [12] 김상형, "Windows API 정복", 가남사
- [13] "OpenGL 프로그래밍 가이드 제3판", 인포북
- [14] Russell Smith, Open Dynamics Engine User Guide